

AD-A114 417 ARMY ELECTRONICS RESEARCH AND DEVELOPMENT COMMAND WS--ETC F/G 4/1
PROGRAM LISTINGS FOR EOSAEL 80-B AND ANCILLARY CODES AGAUS AND --ETC(1)
FEB 82 R G STEINHOFF
UNCLASSIFIED ERADCOM/ASL-TR-0107-V2-SU NL

1 of 6

AD-A114 417

9



-TR-0107

AD

Reports Control Symbol
OSD 1366

2

SUPPLEMENT TO EOSAEL 80
VOLUME II
USER'S MANUAL

PROGRAM LISTINGS FOR EOSAEL 80-B
AND ANCILLARY CODES AGAUS AND FLASH

FEBRUARY 1982

By

R. G. Steinhoff

DTIC FILE COPY

DTIC
ELECTE
MAY 14 1982
S D

Approved for public release; distribution unlimited.



US Army Electronics Research and Development Command
Atmospheric Sciences Laboratory

White Sands Missile Range, NM 88002

82 05 11 016

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

The citation of trade names and names of manufacturers in this report is not to be construed as official Government indorsement or approval of commercial products or services referenced herein.

Disposition

Destroy this report when it is no longer needed. Do not return it to the originator.

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ASL-TR-0107	2. GOVT ACCESSION NO. AD-A114417	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) SUPPLEMENT TO EOSAEL 80 Vol II USER'S MANUAL PROGRAM LISTINGS FOR EOSAEL 80-B AND ANCILLARY CODES AGAUS AND FLASH		5. TYPE OF REPORT & PERIOD COVERED Final Report
7. AUTHOR(s) R. G. Steinhoff		6. PERFORMING ORG. REPORT NUMBER
9. PERFORMING ORGANIZATION NAME AND ADDRESS US Army Atmospheric Sciences Laboratory White Sands Missile Range, NM		8. CONTRACT OR GRANT NUMBER(s)
11. CONTROLLING OFFICE NAME AND ADDRESS US Army Electronics Research and Development Command Adelphi, MD 20783		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS DA Task AH71612111 H710011
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE February 1982
		13. NUMBER OF PAGES 510
		15. SECURITY CLASS. (of this report) UNCLASSIFIED
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		Accession For NTIS GRA&I <input checked="" type="checkbox"/> DTIC TAB <input type="checkbox"/> Unannounced <input type="checkbox"/> Justification
18. SUPPLEMENTARY NOTES		By Distribution/ Availability Dist Spec
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) EOSAEL 80 Computer Codes Transmission Transport and diffusion		A
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) Computer listing of the Electro-Optical Systems Atmospheric Effects Library, version 80-B.		

CONTENTS

INTRODUCTION	7
PROGRAM EOSAEL	9
SUBROUTINE RESET	18
SUBROUTINE COMPLT	19
SUBROUTINE ILLUM	20
FUNCTION SOLARS	21
FUNCTION SMOON	22
FUNCTION JPASCT	23
SUBROUTINE PFUNC	24
SUBROUTINE XSCALE	29
SUBROUTINE SLANT	33
SUBROUTINE TURB	35
FUNCTION DESUB	43
FUNCTION FALPH	44
SUBROUTINE FFT4	45
FUNCTION GAUSS	47
SUBROUTINE MEANYR	48
SUBROUTINE SPECT	49
SUBROUTINE SPREAD	50
SUBROUTINE THETO	51
SUBROUTINE BASCAT	52
SUBROUTINE PFUNC	24
SUBROUTINE BKWD	66
SUBROUTINE CONV	68
SUBROUTINE ELM	69
SUBROUTINE FIND	70
SUBROUTINE FWRD	71
SUBROUTINE GAS	72
SUBROUTINE GMAX	73
SUBROUTINE MATRX	74
SUBROUTINE ROTAT	75
SUBROUTINE SMOOZ	76
SUBROUTINE START	77
SUBROUTINE TRAVRS	79
SUBROUTINE USCA	81
SUBROUTINE SMOKE	82
FUNCTION JPASCT	23
SUBROUTINE CLSMOK	93
SUBROUTINE SCONST	96
SUBROUTINE SMASSP	98
SUBROUTINE STRANS	99
SUBROUTINE WGGEOM	100
FUNCTION QROOT	102
SUBROUTINE XYZINT	103
SUBROUTINE BRATE	108
SUBROUTINE GPUFF	109
SUBROUTINE LZTRAN	111
SUBROUTINE LZIDNM	115
SUBROUTINE DRTRAN	117
SUBROUTINE AMOUNT	130
SUBROUTINE ATMCAL	131
SUBROUTINE AVRG	135
SUBROUTINE CLIMB	136
SUBROUTINE CLOIM	138
SUBROUTINE COMPCL	140
SUBROUTINE CONLEN	143
SUBROUTINE CONVRT	145
FUNCTION CSPHER	147
FUNCTION CWAKE	149
FUNCTION CWINO	151

SUBROUTINE	DIFEQ	154
FUNCTION	DIFFUS	156
FUNCTION	DOTPRD	157
FUNCTION	DTERRI	158
SUBROUTINE	DTERPS	160
SUBROUTINE	DUSTCL	161
FUNCTION	ERF	164
SUBROUTINE	FIT	165
FUNCTION	FUNCT	166
SUBROUTINE	GAMMA	167
FUNCTION	GFUN	168
SUBROUTINE	GRAD2	169
SUBROUTINE	GRAND	170
SUBROUTINE	GREEN	172
SUBROUTINE	GREEN1	173
SUBROUTINE	MOMENT	175
SUBROUTINE	PATH	176
SUBROUTINE	PERP	179
SUBROUTINE	PRECL	180
SUBROUTINE	PRETRN	183
SUBROUTINE	PREVEH	185
SUBROUTINE	RISE	188
SUBROUTINE	RKM	190
SUBROUTINE	SETUP	192
SUBROUTINE	SOURCE	194
SUBROUTINE	TEMP	197
FUNCTION	TMPCAL	198
SUBROUTINE	TRAP	200
SUBROUTINE	TRNCAL	203
SUBROUTINE	TRNCHK	207
SUBROUTINE	TRNCLD	208
SUBROUTINE	UNIT	210
SUBROUTINE	VEHCL	211
SUBROUTINE	VEHTRN	213
SUBROUTINE	VSRC	215
SUBROUTINE	VSUM	217
SUBROUTINE	WIN	218
FUNCTION	WNDCL	219
SUBROUTINE	NMMW	221
FUNCTION	AB	224
FUNCTION	DOP	225
SUBROUTINE	INTRP	226
SUBROUTINE	MMH20	227
SUBROUTINE	MMIDX	228
SUBROUTINE	MMOXY	229
SUBROUTINE	MMRAN	230
SUBROUTINE	MMSNO	232
SUBROUTINE	MMWFG	234
SUBROUTINE	MMWGS	235
FUNCTION	PFR	237
FUNCTION	PSAT	238
FUNCTION	SUPK	239
SUBROUTINE	CLTRAN	240
SUBROUTINE	LAYRXY	244
SUBROUTINE	CYLXY	245
SUBROUTINE	CLEXTN	247
SUBROUTINE	LISOUT	248
SUBROUTINE	DEFSET	250
SUBROUTINE	CLREAD	251
SUBROUTINE	SCREEN	253
SUBROUTINE	XSCALE	259
SUBROUTINE	SLANT	263
FUNCTION	JPASCT	263
SUBROUTINE	PFUNC	264
SUBROUTINE	CMIC	264
SUBROUTINE	CMIC1	261
SUBROUTINE	CMIC3	264

SUBROUTINE	CWIC4	265
SUBROUTINE	ITAM	267
SUBROUTINE	CINV	274
SUBROUTINE	INTAL	279
SUBROUTINE	CASE1	282
SUBROUTINE	CASE2	283
SUBROUTINE	CASE3	284
SUBROUTINE	CYCLE	285
SUBROUTINE	TREQ	286
SUBROUTINE	FCLLOUD	287
SUBROUTINE	ILLUM	20
FUNCTION	SOLARS	21
FUNCTION	SMOON	22
SUBROUTINE	THRMCL	293
SUBROUTINE	SSCLD	294
SUBROUTINE	PFNN	295
SUBROUTINE	MSCLD	297
FUNCTION	ETAINT	298
SUBROUTINE	OVRCST	299
SUBROUTINE	ILLUM	20
FUNCTION	SOLARS	21
FUNCTION	SMOON	22
FUNCTION	G2	303
FUNCTION	E1	304
SUBROUTINE	GRNADE	305
SUBROUTINE	CONCH	309
SUBROUTINE	GGET	311
SUBROUTINE	PARMS	312
SUBROUTINE	EXTIN	313
SUBROUTINE	UMEAN	314
SUBROUTINE	SUMA	315
SUBROUTINE	LGCAT	316
SUBROUTINE	DATRD	317
SUBROUTINE	LT4M	321
SUBROUTINE	XSCALE	29
SUBROUTINE	SLANT	33
SUBROUTINE	ABSORB	338
SUBROUTINE	CKER	336
SUBROUTINE	FREQSL	337
SUBROUTINE	H2OVAP	339
SUBROUTINE	H2O410	340
SUBROUTINE	LTPATH	341
SUBROUTINE	MOLSCF	343
SUBROUTINE	NH3	344
SUBROUTINE	NITRIC	345
SUBROUTINE	NITRO	346
SUBROUTINE	NO2	347
SUBROUTINE	OZONE	348
SUBROUTINE	POINT	349
FUNCTION	RESFN	350
SUBROUTINE	SO2	351
SUBROUTINE	UNIMIX	352
SUBROUTINE	UVOZNE	353
SUBROUTINE	SPOT	354
SUBROUTINE	LT4M	321
SUBROUTINE	XSCALE	29
SUBROUTINE	SLANT	33
SUBROUTINE	PFUNC	34
FUNCTION	SOLARS	21
FUNCTION	SMOON	22
FUNCTION	ALBEDO	361
FUNCTION	BLACK	362
SUBROUTINE	COEFS	363
SUBROUTINE	DIAG	364
SUBROUTINE	INDAT	366

SUBROUTINE	OUTPUT	369
SUBROUTINE	PATHRD	372
SUBROUTINE	ZERO	374
SUBROUTINE	CLIMAT	375
PROGRAM	AGAUS	379
SUBROUTINE	AGXP1	385
SUBROUTINE	AGXP2	390
SUBROUTINE	AGXP3	396
SUBROUTINE	ANGLE	398
FUNCTION	GAMMA	399
SUBROUTINE	GUSET	400
SUBROUTINE	MIEGX	401
SUBROUTINE	WATER	405
SUBROUTINE	GAUS	408
SUBROUTINE	DIMER	409
BLOCK DATA		410
PROGRAM	FLASH1	411
SUBROUTINE	FLASH	412
SUBROUTINE	DATRD	414
SUBROUTINE	DATWT	416
SUBROUTINE	GETIM	418
SUBROUTINE	IRBLC	421
SUBROUTINE	VSBLC	422
EOSAEL TEST DATA (NEWRUN)		423
EOSAEL OUTPUT (EOOUT)		426

INTRODUCTION

This listing of EOSAEL 80-B is a supplement to Volume II¹ and supersedes all previous listings.² The current listing is complete as of 8 February 1982 and has revisions one through five incorporated into it.

EOSAEL 80-B differs from EOSAEL 80 in that modules SPOT, LT4M, NMMW, CLIMAT, BASCAT, and TURB have been extensively revised and, therefore, appear with new sequence numbers. All other modules have their original sequencing, except where revisions have been inserted or deleted.

The programs are listed by module with each module followed by its subroutines. Subroutines that have been listed for prior modules in the listing are not repeated in the source listing. The table of contents lists each module along with all its corresponding subroutines and the page number of each subroutine in the listing. The elements EOMAIN, COMPLT, and RESET, which are always to be resident, appear only at the beginning of the table of contents and the source listing.

Also included herein is a sample input file, NEWRUN, and an output file, EOOUT, produced by using the aforementioned sample input file.

The supplemental codes AGAUS and FLASH are supplied with EOSAEL 80. FLASH is described in appendix A of volume II of the User's Manual¹ and is further described in the comments of the source listing. Operating instructions for the AGAUS code may be found in the comments of the source listing. Manuals for AGAUS are available upon written request from the US Army Atmospheric Sciences Laboratory, White Sands Missile Range, New Mexico.

¹Shirkey, R. C., and S. G. O'Brien, EOSAEL 80, Volume II, User's Manual, ASL-TR-0073, US Army Atmospheric Sciences Laboratory, White Sands Missile Range, NM, 1981.

²Steinhoff, R. G., Program Listings for EOSAEL 80 and Ancillary Codes AGAUS and FLASH, ASL-TR-0073 (Supplement), US Army Atmospheric Sciences Laboratory, White Sands Missile Range, NM, 1981.

```

C      PROGRAM EOSAEL
C      MAIN PROGRAM FOR EOSAEL 80
C
C      REAL LOTRNS,LZTRN,LZTRN,MMTRAN,MMWTRN,IPNAM,IAL,IALB1,IALB2
C      LOGICAL ISPOT,N16,LOREAD
C      COMMON /SPOTLO/ISPOT,LOREAD,N16
C      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK
C      COMMON /CLYMAT/TEMP,PRESS,RH,AH,DP,VIS,CLDAMT,CLDHYT,FOGPRB,
C      1 WNDVEL,WNDDIR,IPASCT
C      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
C      COMMON /GEOMET/PTS(15),IGEOSW
C
C      IOIN - CARD READER
C      IOOUT - PRINTER
C      IPHFUN - UNIT UPON WHICH PHASE FUNCTION DATA RESIDES
C      LOUNIT - UNIT UPON WHICH LT4M ATM DATA RESIDES
C      NDIRTU - UNIT UPON WHICH DRTN DATA RESIDES
C      NCLIMT - UNIT FOR CLIMATOLOGICAL DATA
C      KSTOR - AUXILLARY START/RESTART UNIT FOR BASCAT
C      NPLOTU - OPTIONAL UNIT FOR WRITING RESULTS FOR SUBSEQUENT
C      PLOTTING PURPOSES BY THE USER.
C
C      DIMENSION TRAN(16),RADAC(16),RADG(16),IPROGN(20)
C      DIMENSION IDOPGM(20),IPNAM(40)
C      DIMENSION IAL(12),DAT(10)
C      FOR UNIVAC
C      DATA IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR/
C      1 5,6,3,8,7,24,25/
C      PROGRAM NAMES
C      DATA IPNAM,4HSPOT,4HTURB,4HBASC,4HLT4M,4HXSCA,4HSMOK,4HDRTR,
C      1 4HLZTR,4HNMW,4HCLTR,4HSCRE,4HFCLD,4HOVRC,4HGRNA,4H* ,
C      2 4H* ,4H* ,4H* ,4H* ,4HCLIM,4H ,4H ,4HAT ,
C      3 4H ,4HLE ,4HE ,4HAN ,4HAN ,4H ,4HAN ,4HEN ,
C      4 4HUD ,4HST ,4HDE ,4H ,4H ,4H ,4H ,4H ,
C      5 4HATE /
C      CARD MNEMONICS
C      DATA IAL,4HEGRU,4HVIS ,4HFREQ,4HWAFL,4HWVNU,4HRESF,
C      14HTARG,4HRCVR,4HDESG,4HOBVS,4HBFL,4HGO /
C      DATA PI,TORRMB,CDEGK/3.14159265,1.33322,273.16/
C      DATA PTS/15*0.0/
C      ISTART=0
C      CLDAMT=0.
C      CLDHYT=0.
C      FOGPRB=0.
C      PI2=PI/2.
C      PIRAD=PI/180.
C      TWOPI=2.0*PI
C      WRITE (IOOUT,1060)
C***** I/O *****
C
C** INPUT TO EOSAEL IS CARD ORDER-INDEPENDENT, WITH EACH INPUT RECORD
C** HAVING A FOUR-LETTER IDENTIFIER IN COLUMNS 1-4. THE ONLY EXCEPTION
C** TO THIS RULE IS THE GO SENTINEL CARD, WHICH MUST BE THE LAST RECORD
C** IN THE INPUT SEQUENCE. ALL RECORDS ARE READ IN UNDER THE
C** FORMAT (2A4,1X,10E7.4). INTEGERS MUST BE INPUT AS REAL
C** NUMBERS IN THIS COMMON FORMAT SCHEME. THEY ARE LATER FIXED TO
C** THE INTEGER TYPE. THE IDENTIFIERS FOR EACH OF THE INPUT
C** RECORDS ARE AS FOLLOWS :
C
C-----
C      CARD IDENTIFIER : EORUN
C      VARIABLES READ : NUMRUN
C      NUMRUN - NUMBER OF TIMES EOSAEL DRIVER IS TO BE CYCLED
C      DEFAULT IS 1.
C-----
C      CARD IDENTIFIER : VIS
C      VARIABLES READ : VIS,EXTN55,EXTN
C      VIS - VISIBILITY AT WAVELENGTH OF 0.55 MICRONS (KM)
C      EXTN55 - EXTINCTION COEFFICIENT AT 0.55 MICRONS (KM**-1)
C      EXTN - EXTINCTION COEFFICIENT AT INPUT WAVELENGTH (KM**-1)
C** NOTE : IF THE VIS CARD IS NOT INPUT, A WARNING IS PRINTED
C          AND THE VISIBILITY IS SET TO A DEFAULT VALUE OF 10 KM.

```

C ** NOTE : IF EXTN55 IS INPUT AS A VALUE LESS THAN 0.0001, IT
 C IS SET EQUAL TO THE QUOTIENT 3.912/VIS. IF VIS IS
 C INPUT AS A VALUE LESS THAN 0.0001, IT IS SET EQUAL TO
 C THE QUOTIENT 3.912/EXTN55.
 C ** NOTE : EXTN IS NEEDED ONLY FOR BASCAT

C ** ONLY ONE OF THE FOLLOWING THREE CARDS MAY BE INPUT FOR A GIVEN
 C ** CYCLE OF EOSAEL. IF NONE OF THESE CARDS IS PRESENT, AN ERROR
 C ** MESSAGE IS PRINTED AND EXECUTION IS TERMINATED.

C CARD IDENTIFIER : FREQ
 C VARIABLES READ : FREQ1, FREQ2, MULDV
 C FREQ1 - LOWER INPUT FREQUENCY (GHZ)
 C FREQ2 - HIGHER INPUT FREQUENCY (GHZ)
 C MULDV - FREQUENCY INCREMENT FOR SPOT AND/OR LT4M (GHZ)

C CARD IDENTIFIER : WAVL
 C VARIABLES READ : WAVE1, WAVE2, MULDV
 C WAVE1 - SHORTER INPUT WAVELENGTH (MICRONS)
 C WAVE2 - LONGER INPUT WAVELENGTH (MICRONS)
 C MULDV - WAVELENGTH INCREMENT FOR SPOT AND/OR LT4M (MICRONS)

C CARD IDENTIFIER : WVNUM
 C VARIABLES READ : WVNUM1, WVNUM2, MULDV
 C WVNUM1 - LOWER INPUT WAVENUMBER (CM**-1)
 C WVNUM2 - HIGHER INPUT WAVENUMBER (CM**-1)
 C MULDV - WAVENUMBER INCREMENT FOR SPOT AND/OR LT4M (CM**-1)

C ** THE NEXT CARD DETERMINES WHETHER A SENSOR RESPONSE FUNCTION
 C ** IS DESIRED FOR BROAD BAND CALCULATIONS. THIS OPTION IS
 C ** INVOKED ONLY IF THIS CARD IS PRESENT.

C CARD IDENTIFIER : RESF
 C VARIABLES READ : NONE HERE - SEE SPOT OR LT4M WRITEUP FOR
 C PROPER PLACEMENT OF RESPONSE FUN CARDS.

C ** THE NEXT FIVE CARDS COMPRISE THE GEOMETRICAL OPTION OF EOSAEL.
 C ** THIS OPTION IS USEFUL FOR EOSAEL RUNS WHERE SEVERAL MODULES
 C ** EXAMINE DIFFERENT ATMOSPHERIC OBSCURATION EFFECTS ALONG THE
 C ** SAME PHYSICAL PATH. THE GEOMETRICAL OPTION ASSURES THAT THE
 C ** POINTS OF REFERENCE IN THE SCENARIO UNDER EXAMINATION ARE
 C ** CONSISTENTLY SPECIFIED FOR ALL MODULES. IT SHOULD BE NOTED
 C ** THAT THIS OPTION IS ACTIVATED WHENEVER ANY OF THE FIVE CARDS IS
 C ** ENCOUNTERED. ONCE THE OPTION IS ACTIVATED IT IS IMPORTANT THAT
 C ** AT LEAST THE FIRST TWO CARDS (TARG AND RCVR) BE INPUT TO
 C ** DEFINE THE PHYSICAL PATH, SINCE THIS OPTION WILL OVERRIDE
 C ** POSITIONS OR LENGTHS CONTAINED IN NORMAL INPUT TO ALL MODULES.
 C ** THE GEOMETRICAL INPUT CONSISTS OF FIVE SETS OF COORDINATES
 C ** WHICH OBEY THE FOLLOWING CONVENTIONS :
 C ** (A) ALL COORDINATES ARE DIMENSIONED IN KILOMETERS
 C ** (B) THE Z-AXIS IS POSITIVE UPWARD
 C ** (C) THE Y-AXIS POINTS NORTH
 C ** (D) THE X-AXIS POINTS EAST

C ** THE FIVE GEOMETRICAL CARDS ARE AS FOLLOWS :

C CARD IDENTIFIER : TARG
 C VARIABLES READ : PTS(1),PTS(2),PTS(3)
 C PTS(1-3) - COORDINATES OF THE TARGET (FOR THE DRTRAN MODULE,
 C THESE ARE THE COORDINATES OF THE TRANSMITTER).

C CARD IDENTIFIER : RCVR
 C VARIABLES READ : PTS(4),PTS(5),PTS(6)
 C PTS(4-6) - COORDINATES OF THE RECEIVER OR SEEKER

EOM00680
 EOM00690
 EOM00700
 EOM00710
 EOM00720
 EOM00730
 EOM00740
 EOM00750
 EOM00760
 EOM00770
 EOM00780
 EOM00790
 EOM00800
 EOM00810
 EOM00820
 EOM00830
 EOM00840
 EOM00850
 EOM00860
 EOM00870
 EOM00880
 EOM00890
 EOM00900
 EOM00910
 EOM00920
 EOM00930
 EOM00940
 EOM00950
 EOM00960
 EOM00970
 EOM00980
 EOM00990
 EOM01000
 EOM01010
 EOM01020
 EOM01030
 EOM01040
 EOM01050
 EOM01060
 EOM01070
 EOM01080
 EOM01090
 EOM01100
 EOM01110
 EOM01120
 EOM01130
 EOM01140
 EOM01150
 EOM01160
 EOM01170
 EOM01180
 EOM01190
 EOM01200
 EOM01210
 EOM01220
 EOM01230
 EOM01240
 EOM01250
 EOM01260
 EOM01270
 EOM01280
 EOM01290
 EOM01300
 EOM01310
 EOM01320
 EOM01330
 EOM01340
 EOM01350
 EOM01360
 EOM01370

EOM01990
EOM02000
EOM02010
EOM02020
EOM02030
EOM02040
EOM02050
EOM02060
EOM02070
EOM02080
EOM02090
EOM02100
EOM02110
EOM02120
EOM02130
EOM02140
EOM02150
EOM02160
EOM02170
EOM02180
EOM02190
EOM02200
EOM02210
EOM02220
EOM02230
EOM02240
EOM02250
EOM02260
EOM02270
EOM02280
EOM02290
EOM02300
* EOM02310
EOM02320
EOM02330
EOM02340
EOM02350
EOM02360
EOM02370
EOM02380
EOM02390
EOM02400
EOM02410
EOM02420
EOM02430
EOM02440
EOM02450
EOM02460
EOM02470
EOM02480
EOM02490
EOM02500
EOM02510
EOM02520
EOM02530
EOM02540
EOM02550
EOM02560
EOM02570
EOM02580
EOM02590
EOM02600
EOM02610
EOM02620
EOM02630
EOM02640
EOM02650
EOM02660
EOM02670
EOM02680

CARD IDENTIFIER : XSEE
VARIABLES READ : IDOFGM

IDOPGM - NUMBER OF TIMES THE SELECTED MODULE IS TO
 BE CYCLED WITHIN EACH EOSAEL CYCLE - DEFAULT IS ONE.

NO.	MODULE IDENTIFIER	WAVELENGTH	RANGES	FREQ(GHZ)
1	SPOT	.25-2.,3.-5.,8.-12.*		
2	TURB	LT 14.		
3	BASCAT	ANY WAVELENGTH IN DATA	FILE	IPHFUN
4	LT4M	.25-2.,3.-5.,8.-12.*		
5	XSCALE	1.06,3-5,8-12*		
6	SMOKE	.4-1.2,3-5,8-12*		94.
7	DRTRAN	.4-1.1,3.5-4.,8.5-12.*		94.-140.
8	LZTRAN	.8-11.		
9	NMMW			10-350
10	CLTRAN	.2-2.,3.-5.,8.-12.*		
11	SCREEN	N/A		
12	FCLLOUD	ANY WAVELENGTH IN DATA	FILE	IPHFUN
13	OVRCS	ANY WAVELENGTH		
14	GRNADE	SAME AS SMOKE		

```

C** NOTE : THE DATA SPECIFIC TO EACH MODULE MUST BE INPUT IN
C          THE SEQUENCE IN THE ABOVE LIST.

```

CARD IDENTIFIER : GU
VARIABLES READ : NONE

END OF READ SENTINEL (MUST BE LAST CARD READ).

```
C*****
NUMRUN=1
IRFLAG=0
READ (IGIN,1000) IALB1,IALB2,(DAT(L),L=1,10)
IF (IALB1.NE.IAL(1)) GO TO 10
NUMRUN=IFIX(DAT(1))
IF (NUMRUN.EQ.0) NUMRUN=1
GO TO 20
C   SET FLAG IF EORUN IS NOT THE FIRST CARD
10  IRFLAG=1
20  CONTINUE
DO 580 JRUN=1,NUMRUN
C   INITIALIZATION
DO 30 I=1,20
C   PROGRAM SELECTOR
IPROGN(I)=0
C   PROGRAM CYCLE DEFAULT
30  IDOPGM(I)=1
C   TRANSMISSIONS
LOTRNS=1.
XSTRN=1.
SMKTRN=1.
DRTRN=1.
LZTRN=1.
MMWTRN=1.
CLTRN=1.0
GRNTRN=1.0
IF (JRUN.GT.1) WRITE (IOOUT,1070) JRUN
C   FREQUENCY, WAVELENGTH, WAVENUMBER INDICATOR
IFW=0
C   GEOMETRICAL OPTION SWITCH
IGEOSW=0
C   SENSOR RESPONSE FUNCTION OPTION SWITCH
NR=0
VIS=0.0
EXTN55=0.0
EXTN=0.0
DO 220 J=1,25
```

	IF (IRFLAG.EQ.1) WRITE (IOOUT,1010)	EOM02690
C	SUPPRESS READ IN CASE FIRST CARD PREVIOUSLY READ WASNT EORUN.	EOM02700
	IF (IRFLAG.EQ.0) READ (IOIN,1000) IALB1,IALB2,(DAT(L),L=1,10)	EOM02710
	IRFLAG=0	EOM02720
	INOPT=0	EOM02730
	IF (J.EQ.25) GO TO 230	EOM02740
	DO 40 KK=1,12	EOM02750
C	CHECK FOR CARD TYPES, NOT PROGRAM SELECTOR	EOM02760
	IF (IALB1.NE.IAL(KK)) GO TO 40	EOM02770
	INOPT=KK	EOM02780
	IF (INOPT.GE.3.AND.INOPT.LE.5) IFW=INOPT	EOM02790
C	GO CARD FOUND	EOM02800
	IF (INOPT.EQ.12) GO TO 250	EOM02810
	GO TO 80	EOM02820
40	CONTINUE	EOM02830
C	SEARCH FOR PROGRAMS HERE	EOM02840
	DO 50 KK=1,20	EOM02850
	IF (IALB1.NE.IPNAM(KK)) GO TO 50	EOM02860
	IPROGN(KK)=KK	EOM02870
	IF (DAT(1).GT.1.0) IDOPGM(KK)=IFIX(DAT(1))	EOM02880
	IF (KK.EQ.20) GO TO 60	EOM02890
	GO TO 220	EOM02900
50	CONTINUE	EOM02910
	GO TO 240	EOM 2920
C	CLIMATOLOGICAL OPTION INVOKED	EOM02930
60	ICLMAT=IFIX(DAT(1))	EOM02940
	IF (ICLMAT.EQ.2) GO TO 70	EOM02950
	LOCAT=IFIX(DAT(2))	EOM02960
	MONTH=IFIX(DAT(3))	EOM02970
	NHOUR=IFIX(DAT(4))	EOM02980
	IWIND=IFIX(DAT(5))	EOM02990
	NPRT=IFIX(DAT(6))	EOM03000
	GO TO 220	EOM03010
70	IPASCT=IFIX(DAT(2))	EOM03020
	TEMP=DAT(3)	EOM03030
	PRESS=DAT(4)	EOM03040
	RH=DAT(5)	EOM03050
	AH=DAT(6)	EOM03060
	DP=DAT(7)	EOM03070
	VIS=DAT(8)	EOM03080
	WINDVEL=DAT(9)	EOM03090
	WINDDIR=DAT(10)	EOM03100
	GO TO 220	EOM03110
C	GEOMETRICAL OPTION INVOKED	EOM03120
80	IF (INOPT.GT.6) GO TO 90	EOM03130
C	CARD SETUP SWITCHING	EOM03140
	GO TO (210,160,170,180,190,200),INOPT	EOM03150
90	LPTSSW=INOPT-6	EOM03160
	IGEOSW=1	EOM03170
	DO 150 K=1,3	EOM03180
	GO TO (100,110,120,130,140),LPTSSW	EOM03190
100	PTS(K)=DAT(K)	EOM03200
	GO TO 150	EOM03210
110	PTS(K+3)=DAT(K)	EOM03220
	GO TO 150	EOM03230
120	PTS(K+6)=DAT(K)	EOM03240
	GO TO 150	EOM03250
130	PTS(K+9)=DAT(K)	EOM03260
	GO TO 150	EOM03270
140	PTS(K+12)=DAT(K)	EOM03280
150	CONTINUE	EOM03290
	GO TO 220	EOM03300
C	VISIBILITY CARD	EOM03310
160	VIS=DAT(1)	EOM03320
	EXTN55=DAT(2)	EOM03330
	EXTN=DAT(3)	EOM03340
	GO TO 220	EOM03350
C	FREQUENCY CARD	EOM03360
170	FREQ1=DAT(1)	EOM03370
	FREQ2=DAT(2)	EOM03380

```

      MULDV=IFIX(DAT(3))
      GO TO 220
C 180 WAVELENGTH CARD
      WAVE1=DAT(1)
      WAVE2=DAT(2)
      MULDV=IFIX(DAT(3))
      GO TO 220
C 190 WAVENUMBER CARD
      WVNUM1=DAT(1)
      WVNUM2=DAT(2)
      MULDV=IFIX(DAT(3))
      GO TO 220
C 200 SENSOR OPTION INVOKED
      NR=1
      GO TO 220
      WRITE (IOOUT,1010)
      CONTINUE
C 210 IF (IFW.NE.0) GO TO 250
      ERROR CHECK ON WAVENUMBER, WAVELENGTH, OR FREQUENCY
      WRITE (IOOUT,1020)
      GO TO 580
C 230 WRITE (IOOUT,1030)
      GO TO 250
C 240 UNKNOWN CARD TYPE
      WRITE (IOOUT,1040) IALB1,IALB2
C 250 CONTINUE
      SELECT FREQUENCY, WAVELENGTH, OR WAVENUMBER
      IF (IFW-4) 260,270,280
C 260 WVNUM1=FREQ1/30.
      WVNUM2=FREQ2/30.
      WAVE1=0.
      IF (FREQ2.GT..0001) WAVE1=3.E+05/FREQ2
      WAVE2=3.E+05/FREQ1
      GO TO 290
C 270 WVNUM1=0.
      IF (WAVE2.GT..0001) WVNUM1=1.E+04/WAVE2
      WVNUM2=1.E+04/WAVE1
      FREQ1=30.*WVNUM1
      FREQ2=30.*WVNUM2
      GO TO 290
C 280 FREQ1=30.*WVNUM1
      FREQ2=30.*WVNUM2
      WAVE1=0.
      IF (WVNUM2.GT..0001) WAVE1=1.E+04/WVNUM2
      WAVE2=1.E+04/WVNUM1
C 290 CONTINUE
      IF (VIS.LT..0001.AND.EXTN55.LT.0001) WRITE (IOOUT,1050)
      IF (VIS.LT..0001.AND.EXTN55.LT.0001) VIS=10.
      IF (EXTN55.GT..0001) VIS=3.912/EXTN55
      IF (VIS.GT..0001) EXTN55=3.912/VIS
C 300 OUTPUT INFORMATION
      WRITE (IOOUT,1080)
      DO 300 I=1,20
      IF (IPROGN(I).EQ.1) WRITE (IOOUT,1090) IPHAM(I),IPHAM(I+20)
      WRITE (IOOUT,1100) WVNUM1,WVNUM2,WAVE1,WAVE2,FREQ1,FREQ2
C 310 CLIMAT USES UNIT NCLIMT
      IF (ICLMAT.EQ.1) CALL CLIMAT(LOCAT,MONTH,NHOUR,IWIND,NPRT,TEMP,
      PRESS,RH,AH,DP,VIS,WNDVEL,WNDDIR,IPASCT)
      WRITE (IOOUT,1110) VIS
      IF (ICLMAT.EQ.2) ICLMAT=1
C***** SPOT CONTRAST PGM *****
      IF (IPROGN(1).NE.1) GO TO 320
      IPGM1=IDOPGM(1)
      WRITE (IOOUT,1130)
      DO 310 I=1,IPGM1
C 310 SPOT USES UNITS: IPHFUN - PHASE FUNCTION; LOUNIT - LT4M DATA
      CALL SPOT(WVNUM1,WVNUM2,VIS,NR,IERR,MULDV)
      CALL RESET(IERR)
C***** TURBULENCE PGM *****
      320 IF (IPROGN(2).NE.2) GO TO 340

```

```

EOM03390
EOM03400
EOM03410
EOM03420
EOM03430
EOM03440
EOM03450
EOM03460
EOM03470
EOM03480
EOM03490
EOM03500
EOM03510
EOM03520
EOM03530
EOM03540
EOM03550
EOM03560
EOM03570
EOM03580
EOM03590
EOM03600
EOM03610
EOM03620
EOM03630
EOM03640
EOM03650
EOM03660
EOM03670
EOM03680
EOM03690
EOM03700
EOM03710
EOM03720
EOM03730
EOM03740
EOM03750
EOM03760
EOM03770
EOM03780
EOM03790
EOM03800
EOM03810
EOM03820
EOM03830
EOM03840
EOM03850
EOM03860
EOM03870
EOM03880
EOM03890
EOM03900
EOM03910
EOM03920
EOM03930
EOM03940
EOM03950
EOM03960
EOM03970
EOM03980
EOM03990
EOM04000
EOM04010
EOM04020
EOM04030
EOM04040
EOM04050
EOM04060
EOM04070
EOM04080

```

```

      IPGM2=IDOPGM(2)
      WRITE (IOOUT,1140)
      DO 330 I=1,IPGM2
330    CALL TURB(WAVE1,IERR)
      CALL RESET(IERR)
C***** LASER MULTIPLE SCATTERING PGM *****
340 IF (IPROGN(3).NE.3) GO TO 360
      IPGM3=IDOPGM(3)
      ISFOT=.FALSE.
      WRITE (IOOUT,1150)
      DO 350 I=1,IPGM3
C    BASCAT USES UNIT IPHFUN FOR PHASE FUNCTION DATA
350    CALL BASCAT(WAVE1,EXTN,IERR)
      CALL RESET(IERR)
C***** LT4M PGM *****
360 IF (IPROGN(4).NE.4) GO TO 380
C    LT4M READS ATM DATA FROM LOUNIT
      ISPOT=.FALSE.
      LOREAD=.TRUE.
      IPGM4=IDOPGM(4)
      DO 370 I=1,IPGM4
        CALL LT4M(H1,H2,ANGLE,ITYPE,IXY,TRAN,RADA,RADG,IEMISS,LEN,MODEL,
        VIS,WVNUM1,WVNUM2,T1,ICLMAT,IERR,NR,HAZE,MULDV)
370    LOTRNS=LOTRNS*TRAN(1)
      CALL RESET(IERR)
C***** XSCALE EXTINCTION PGM *****
380 IF (IPROGN(5).NE.5) GO TO 400
      WRITE (IOOUT,1160)
      IPGM5=IDOPGM(5)
      DO 390 I=1,IPGM5
        CALL XSCALE(WAVE1,VIS,EXTN55,XTRN,IERR,0,0,0.,0.)
390    XSTRN=XSTRN*XTRN
      CALL RESET(IERR)
C***** SMOKE PGM *****
400 IF (IPROGN(6).NE.6) GO TO 420
      WRITE (IOOUT,1170)
      IPGM6=IDOPGM(6)
      DO 410 N=1,IPGM6
        CALL SMOKE(WAVE1,ICLMAT,STRANS,IERR)
410    SMKTRN=SMKTRN*STRANS
      CALL RESET(IERR)
C***** DRTRAN PGM *****
420 IF (IPROGN(7).NE.7) GO TO 440
      WRITE (IOOUT,1180)
      IPGM7=IDOPGM(7)
      HOLDWV=WAVE1
      IF (IFW.EQ.1) WAVE1=WAVE2
      DO 430 N=1,IPGM7
C    DRTRAN USES NDIRTU FOR DATA
        IF (N.GT.1) WRITE (IOOUT,1120)
        CALL DRTRAN(WAVE1,ICLMAT,TRNLOS,IERR)
430    DRTRN=DRTRN*TRNLOS
      WAVE1=HOLDWV
      CALL RESET(IERR)
C***** LASER TRANSMISSION PGM *****
440 IF (IPROGN(8).NE.8) GO TO 460
      WRITE (IOOUT,1190)
      IPGM8=IDOPGM(8)
      DO 450 I=1,IPGM8
        CALL LZTRAN(WAVE1,ICLMAT,LAZTRN,IERR)
450    LZTRN=LZTRN*LAZTRN
      CALL RESET(IERR)
C***** NEAR MILLIMETER WAVE PGM *****
460 IF (IPROGN(9).NE.9) GO TO 480
      WRITE (IOOUT,1200)
      IPGM9=IDOPGM(9)
      DO 470 I=1,IPGM9
        CALL MMW(FREQ1,ICLMAT,MMTRAN,IERR)
470    MMWTRN=MMWTRN*MMTRAN
      CALL RESET(IERR)

```

```

EOM04090
EOM04100
EOM04110
EOM04120
EOM04130
EOM04140
EOM04150
EOM04160
EOM04170
EOM04180
EOM04190
EOM04200
EOM04210
EOM04220
EOM04230
EOM04240
EOM04250
EOM04260
EOM04270
EOM04280
EOM04290
EOM04300
EOM04310
EOM04320
EOM04330
EOM04340
EOM04350
EOM04360
EOM04370
EOM04380
EOM04390
EOM04400
EOM04410
EOM04420
EOM04430
EOM04440
EOM04450
EOM04460
EOM04470
EOM04480
EOM04490
EOM04500
EOM04510
EOM04520
EOM04530
EOM04540
EOM04550
EOM04560
EOM04570
EOM04580
EOM04590
EOM04600
EOM04610
EOM04620
EOM04630
EOM04640
EOM04650
EOM04660
EOM04670
EOM04680
EOM04690
EOM04700
EOM04710
EOM04720
EOM04730
EOM04740
EOM04750
EOM04760
EOM04770
EOM04780

```

```

C***** CLOUD TRANSMISSION PGM *****EOM04790
480 IF (IPROGN(10).NE.10) GO TO 500EOM04800
      IPGM10=IDOPGM(10)EOM04810
      DO 490 I=1,IPGM10EOM04820
        WRITE (IOOUT,1210)EOM04830
        CALL CLTRAN(CTRANS,WAVE1,I,IERR)EOM04840
      CLTRN=CLTRN*CTRANS EOM04850
      CALL RESET(IERR)EOM04860
C**** CWIC MUNITION EXPENDITURES/INVERSE STATIC TARGET DETECTION PGM ***EOM04870
500 IF (IPROGN(11).NE.11) GO TO 520EOM04880
      WRITE (IOOUT,1220)EOM04890
      IPGM11=IDOPGM(11)EOM04900
      DO 510 I=1,IPGM11EOM04910
        CALL SCREEN(IERR,ICLMAT)EOM04920
      CALL RESET(IERR)EOM04930
C***** FINITE CLOUD RADIATIVE TRANSFER PGM *****EOM04940
520 IF (IPROGN(12).NE.12) GO TO 540EOM04950
      WRITE (IOOUT,1230)EOM04960
      IPGM12=IDOPGM(12)EOM04970
C      FCLCLOUD USES IPHFUN FOR PHASE FUNCTION DATAEOM04980
      DO 530 I=1,IPGM12EOM04990
        CALL FCLCLOUD(WAVE1,FTRANS,IERR)EOM05000
      CALL RESET(IERR)EOM05010
C***** OVERCAST SKY RADIATIVE TRANSFER PGM *****EOM05020
540 IF (IPROGN(13).NE.13) GO TO 560EOM05030
      WRITE (IOOUT,1240)EOM05040
      IPGM13=IDOPGM(13)EOM05050
      DO 550 I=1,IPGM13EOM05060
        CALL OVRCAST(WAVE1,OTRANS,IERR)EOM05070
      CALL RESET(IERR)EOM05080
C***** SELF-SCREENING SMOKE GRENADE PGM *****EOM05090
560 IF (IPROGN(14).NE.14) GO TO 575EOM05100
      WRITE (IOOUT,1250)EOM05110
      IPGM14=IDOPGM(14)EOM05120
      DO 570 I=1,IPGM14EOM05130
        CALL GRNADE(WAVE1,ICLMAT,GRTRAN,IERR)EOM05140
      GRNTRN=GRNTRN*GRTRAN EOM05150
      CALL RESET(IERR)EOM05160
C*****EOM05170
575 IF (IPROGN(4).GT.0.OR. IPROGN(5).GT.0.OR. IPROGN(6).GT.0.OR. IPROGN(7)
+ .GT.0.OR. IPROGN(8).GT.0.OR. IPROGN(9).GT.0.OR. IPROGN(10).GT.0.OR.
+ IPROGN(14).GT.0) GO TO 576
      GO TO 580
576 CALL COMPLT(LOTRNS,XSTRN,SMKTRN,DRTN,LZTRN,MMWTRN,GRNTRN,CLTRN)EOM05180
580 CONTINUEEOM05190
      WRITE (IOOUT,1260)EOM05200
      STOPEOM05210
C      EOM05220
C      EOM05230
1000 FORMAT(2A4,1X,10E7.4)EOM05240
1010 FORMAT(1H0,20X,75H***EOMAIN WARNING*** EORUN CYCLE CARD OUT OF SEQUEEOM05250
      UENCE, DEFAULT TO ONE CYCLE //)EOM05260
1020 FORMAT(1H0,20X,74H***EOMAIN ERROR*** FREQ, WAVL, OR WYNUM CARD WASEOM05270
      NOT INPUT, RUN TERMINATED //)EOM05280
1030 FORMAT(1H0,20X,46H***EOMAIN ERROR*** END OF READ SENTINEL ABSENT /EOM05290
      1//,1X,20X,28HRESULTS MAY BE UNPREDICTABLE)EOM05300
1040 FORMAT(1H0,20X,80H***EOMAIN ERROR*** INPUT CARD DETECTED WHICH DQEEOM05310
      IS NOT MATCH CORRECT INPUT FORMAT//,1X,20X,13HTHE CARD WAS:,2X,2A4)EOM05320
1050 FORMAT(1H0,20X,24H*** EOSAEL WARNING ****,/,1X,20X,15HVISIBILITYEOM05330
      1 AND 47HEXTINCTION = 0.0, VISIBILITY CHANGED TO 10.0 KM//)EOM05340
1060 FORMAT(1H1,//////,1X,50X,30(1H*),/,1X,50X,1H*,28X,1H*,/EOM05350
      1,1X,50X,30H* ELECTRO-OPTICAL SYSTEMS *,/,1X,50X,1H*,EOM05360
      2,28X,1H*,/,1X,50X,30H* ATMOSPHERIC EFFECTS *,/EOM05370
      3,1X,50X,1H*,28X,1H*,/,1X,50X,25H* LIBRARYEOM05380
      4,5H *,/,1X,50X,1H*,28X,1H*,/,1X,50X,30(1H*))EOM05390
1070 FORMAT(1H1,//////,58X,11HRUN NUMBER ,12)EOM05400
1080 FORMAT(///,1X,51X,28HINDIVIDUAL MODULES SELECTED)EOM05410
1090 FORMAT(1X,62X,2A4)EOM05420
1100 FORMAT(1H0,63X,9HBEGINNING,12X,6HENDING,/,/,39X,14HWAVENUMBER(CM*EOM05430
      1,4H*-1),6X,F10.3,10X,F10.3,/,/,39X,19HWAVELENGTH(MICRONS),EOM05440

```

```

2          5X,F10.3,10X,F10.3,/,39X,14HFREQUENCY(GHZ),5X,F15.3,5X, EOM05450
3          F15.3,/, EOM05460
1110 FORMAT (1H0,62X,10HVISIBILITY,/,62X,F5.2,3H KM) EOM05470
1120 FORMAT(1H1) EOM05480
1130 FORMAT(1H1,40X,20HSPOT CONTRAST MODULE ///) EOM05490
1140 FORMAT(1H1,40X,17HTURB LASER MODULE ///) EOM05500
1150 FORMAT(1H1,40X,30HBASCAT LASER SCATTERING MODULE ///) EOM05510
1160 FORMAT (1H1,40X,46HXSCALE HORIZONTAL-SLANT PATH EXTINCTION MODULE EOM05520
      1 ///) EOM05530
1170 FORMAT (1H1,45X,19HSMOKE MODEL MODULE ///) EOM05540
1180 FORMAT (1H1,40X,26H DIRT TRANSMISSION MODULE ///) EOM05550
1190 FORMAT (1H1,40X,28H LASER TRANSMITTANCE MODULE ///) EOM05560
1200 FORMAT (1H1,45X,29H NEAR MILLIMETER WAVE MODULE ///) EOM05570
1210 FORMAT(1H1,40X,27H CLOUD TRANSMITTANCE MODULE ///) EOM05580
1220 FORMAT(1H1,20X,43HCWIC MUNITION EXPENDITURES / INVERSE STATIC EOM05590
      1 24H TARGET DETECTION MODULE ) EOM05600
1230 FORMAT(1H1,40X,38HFINITE CLOUD RADIATIVE TRANSFER MODULE ///) EOM05610
1240 FORMAT(1H1,40X,38HOVERCAST SKY RADIATIVE TRANSFER MODULE ///) EOM05620
1250 FORMAT(1H1,40X,35HSELF-SCREENING SMOKE GRENADE MODULE ///) EOM05630
1260 FORMAT (1X,/,/,1X,50X,14HEND EOSAEL RUN) EOM05640
      END EOM05650

```

```

SUBROUTINE RESET (IERR)
THE PURPOSE OF THIS ROUTINE IS: (1) TO RESET THE SEQUENCING OF
DATA CARDS DUE TO AN ERROR IN A PREVIOUS MODULE OR (2) TO READ
A SENTINAL CARD THAT DELINEATES THE END OF A DATA SET (SEE
BELOW FOR DEFINITION OF A DATA SET) OR (3) TO STOP THE PROGRAM -
THIS LAST MODE IS USALLY FOR DEBUGGING OR TO ONLY CHANGE A CARD
IN A COMPLETE RUN.

TO DELINEATE THE END OF A DATA SET A CARD THAT HAS JUST END
ON IT MUST BE INSERTED AS A SENTINAL CARD: A DATA SET IS
DEFINED AS THAT COMPLETE SET OF CARDS NECESSARY TO RUN THE
CALLED MODULE THE NUMBER OF TIMES AS SPECIFIED ON THE IDOPGM(1)
CARD. STOP MAY ALSO BE INSERTED AS A SENTINAL CARD, IN WHICH
CASE THE PROGRAM WILL BE TERMINATED AT THAT POINT - THIS IS
NOT THE NORMAL TERMINATION.

COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLINT,KSTOR,NPLOTUR
DATA ICHCK1,ICHCK2,ISTOP1,ISTOP2 /2HEN,2HD,2HST,2HOP/
IF(IERR.EQ.1) GO TO 1
CONTINUE
FOR UNIVAC AND IBM
READ (IOIN,100,END=2) ISNTL1,ISNTL2
READ (IOIN,100) ISNTL1,ISNTL2
C TO EXECUTE THIS ROUTINE ON A 'CDC' MACHINE COMMENT OUT THE
C PRECEDING LINE AND UNCOMMENT THE NEXT TWO LINES C3 AND C4.
C3 READ (IOIN,100) ISNTL1,ISNTL2
C4 IF (EOF(IOIN)) 2,10
10 IF ((ISNTL1.NE.ICHCK1.AND.ISNTL2.NE.ICHCK2).AND.
1 (ISNTL1.NE.ISTOP1.AND.ISNTL2.NE.ISTOP2)) GO TO 5
IF (ISNTL1.EQ.ISTOP1.AND.ISNTL2.EQ.ISTOP2) STOP
RETURN
WRITE (IOOUT,102)
FOR UNIVAC AND IBM
C6 READ (IOIN,100,END=2) ISNTL1,ISNTL2
6 READ (IOIN,100) ISNTL1,ISNTL2
C TO EXECUTE THIS ROUTINE ON A 'CDC' MACHINE COMMENT OUT THE
C PRECEDING LINE AND UNCOMMENT THE NEXT TWO LINES C3 AND C4.
C3 READ (IOIN,100) ISNTL1,ISNTL2
C4 IF (EOF(IOIN)) 2,20
20 CONTINUE
IF ((ISNTL1.NE.ICHCK1.AND.ISNTL2.NE.ICHCK2).AND.
1 (ISNTL1.NE.ISTOP1.AND.ISNTL2.NE.ISTOP2)) GO TO 6
IF (ISNTL1.EQ.ISTOP1.AND.ISNTL2.EQ.ISTOP2) STOP
IERR=0
RETURN
2 WRITE (IOOUT,101) IOIN
100 FORMAT (2A2)
101 FORMAT (1X,120(1H*),/,1X,29H ERROR IN INPUT CONTROL FILE ,I4,
+ 21H - PROGRAM TERMINATED.,/,1X,120(1H*))
102 FORMAT(1H0,50H**** CARD SEQUENCE RESET DUE TO ERROR IN PREVIOUS ,
1 15HMODULE (IERR=1)///)
STOP
END

```

```

RES00010
RES00020
RES00030
RES00040
RES00050
RES00060
RES00070
RES00080
RES00090
RES00100
RES00110
RES00120
RES00130
RES00140
RES00150
RES00160
RES00170
RES00180
RES00190
RES00200
RES00210
RES00220
RES00230
RES00240
RES00250
RES00260
RES00270
RES00280
RES00290
RES00300
RES00310
RES00320
RES00330
RES00340
RES00350
RES00360
RES00370
RES00380
RES00390
RES00400
RES00410
RES00420
RES00430
RES00440
RES00450
RES00460
RES00470
RES00480
RES00490
RES00500
RES00510
RES00520
RES00530

```

```

SUBROUTINE COMPLT<LOTRNS,XSTRN,
+      SMKTRN,DRTRN,LZTRN,MMWTRN,GRNTRN,CLTRN>
REAL LOTRNS,LZTRN,MMWTRN
COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
LOWTRAN = LOTRNS
XSCALE = XSTRN
SMOKE = SMKTRN
DRTRAN = DRTRN
LZTRAN = LZTRN
NMMW = MMWTRN
CLTRAN = CLTRN
GRNADE = GRNTRN
TRAN=LOTRNS*XSTRN*SMKTRN*DRTRN*CLTRN*GRNTRN*MMWTRN*LZTRN
WRITE (IOOUT,100) TRAN
100  FORMAT (///1X,20X,24HCOMBINED TRANSMISSION FO,15HR THE SELECTED ,
1  10HMODULES = ,E10.4)
RETURN
END

```

```

COM00010
COM00020
COM00030
COM00040
COM00050
COM00060
COM00070
COM00080
COM00090
COM00100
COM00110
COM00120
COM00130
COM00140
COM00150
COM00160
COM00170
COM00180

```



```

SUBROUTINE ILLUM(LAMBDA,LD,E0) ILL00010
REAL LAMBDA,LUNPHA ILL00020
COMMON/IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU ILL00030
C***** ILL00040
C SUBROUTINE ILLUM RETURNS THE EXTRATERRESTRIAL IRRADIANCE E0 ILL00050
C AT WAVELENGTH LAMBDA. IF LD = 0, THE VALUE GIVEN IS SOLAR ILL00060
C IRRADIANCE. IF 1 < LD < 28 THE VALUE GIVEN IS LUNAR IRRAD- ILL00070
C IANCE ON LUNAR DAY LD, WITH DAY 28 CORRESPONDING TO FULL ILL00080
C MOON AND DAY 14 BEING NEW MOON. ILL00090
C SUBROUTINE COMPUTES VALUE OF LUNAR PHASE ANGLE, IF REQUIRED, ILL00100
C AND CALLS ONE OF THE EOSAEL ROUTINES SOLARS OR SMOON. ILL00110
C***** ILL00120
C IF(ILD.GT.0) GO TO 10 ILL00130
C E0=SOLARS(LAMBDA) ILL00140
C GO TO 100 ILL00150
C LD GT 0 => E0 = LUNAR VALUE ILL00160
C ILL00170
C 10 ILD=LD ILL00180
C IF(ILD.GT.14) ILD=28-ILD ILL00190
C LUNPHA=180.0*FLOAT(ILD)/14.0 ILL00200
C E0=SMOON(LAMBDA,LUNPHA) ILL00210
C 100 WRITE(IOOUT,1000) E0 ILL00220
C RETURN ILL00230
C 1000 FORMAT(32H0 EXTRATERRESTRIAL IRRADIANCE= ,1PE10.4,11H W/M2-SR-MU) ILL00240
C END ILL00250

```


CCCCCCCC

```

FUNCTION SMOON(WLAM,ANGLE)
CALCULATE THE INTENSITY OF MOONLIGHT FOR WAVELENGTH (WLAM)
AND PHASE ANGLE (ANGLE)
UNITS:
    SMOON ... WATTS M-2 MICRON-1
    ANGLE ... DEGREES
    WLAM .... MICRONS
SMOON=0.0
IF (ANGLE.GT.160.) RETURN
SMOON=(3.426E-9*ANGLE**4-1.63E-6*ANGLE**3+3.01E-4*
1 ANGLE**2-.0266*ANGLE+.9882)*100.
ALBED=0.4
IF (WLAM.GE.5.) GO TO 200
IF (WLAM.GT.2.8) GO TO 100
IF (WLAM.LE.1.) ALBED=3.9633*WLAM**4-10.7306*WLAM**3+
1 10.2188*WLAM**2-3.8208*WLAM+.5512
IF (WLAM.GT.1.) ALBED=.0482*WLAM**4-.3283*WLAM**3+
1 .7584*WLAM**2-.5745*WLAM+.2808
GO TO 200
100 ALBED=.350+(.500-.350)*(WLAM-2.8)/2.2
200 SMOON=2.04472E-07*SOLARS(WLAM)*ALBED*SMOON
RETURN
END

```

SMN00010
SMN00020
SMN00030
SMN00040
SMN00050
SMN00060
SMN00070
SMN00080
SMN00090
SMN00100
SMN00110
SMN00120
SMN00130
SMN00140
SMN00150
SMN00160
SMN00170
SMN00180
SMN00190
SMN00200
SMN00210
SMN00220
SMN00230
SMN00240
SMN00250

C	FUNCTION JPASCT(ICAT)	JPA00010
C	THIS FUNCTION CONVERTS THE INTEGER CODE FOR PASQUILL CATEGORY	JPA00020
	TO THE ALPHA CHARACTER	JPA00030
	DIMENSION NPASCT(6)	JPA00040
	DATA NPASCT/1HA,1HB,1HC,1HD,1HE,1HF/	JPA00050
	JPASCT=NPASCT(ICAT)	JPA00060
	RETURN	JPA00070
	END	JPA00080

DO 250 L=1,LMAX	PFU00710
250 ANG(L)=COS(ANG(L)*PIRAD)	PFU00720
C*** REDUCE DUE TO SENTINEL OF 999.99	PFU00730
C	PFU00740
IF(L11.LT.65)LMAX=K-1	PFU00750
IDNM=IDN-1	PFU00760
KMAX=IFIX(ALOG(FLOAT(LMAX-1))/ALOG(2.0)+0.1)	PFU00770
C	PFU00780
C*** RESET PARAMETERS FOR BASCAT PROCESSING IF APPROPRIATE.	PFU00790
C	PFU00800
IF(ISPOT) GO TO 260	PFU00810
NWVL=1	PFU00820
WVL(1)=SINGWV	PFU00830
260 CONTINUE	PFU00840
IF((IDN.EQ.1).OR.(IDN.EQ.0.)) GO TO 1050	PFU00850
C	PFU00860
C*** READ PAST AEROSOL DATA NOT OF CURRENT INTEREST	PFU00870
C	PFU00880
DO 1000 I=1,IDNM	PFU00890
DO 1000 II=1,16	PFU00900
READ(IPHFUN,300) IANG,ID,WAVE,ALBE,BEX,BSC	PFU00910
300 FORMAT(2(I2,1X),F5.2,1X,F8.6,1X,2(E12.6,1X))	PFU00920
IF(IANG.NE.LMAX) GO TO 493	PFU00930
READ(IPHFUN,400) (PF(L),L=1,LMAX)	PFU00940
400 FORMAT(6(E12.6,1X))	PFU00950
1000 CONTINUE	PFU00960
1050 CONTINUE	PFU00970
C	PFU00980
C*** OMIT WAVELENGTH CHECKS FOR USER-DEFINED PHASE FUNCTION.	PFU00990
C	PFU01000
IF(IDN.EQ.0) GO TO 1070	PFU01010
C	PFU01020
C*** THE NEXT LOOP PERFORMS THE FOLLOWING OPERATIONS :	PFU01030
C	PFU01040
C	PFU01050
C	PFU01060
C	PFU01070
C	PFU01080
C	PFU01090
C	PFU01100
C	PFU01110
C	PFU01120
C	PFU01130
C	PFU01140
C	PFU01150
C	PFU01160
C	PFU01170
C	PFU01180
C	PFU01190
C	PFU01200
C	PFU01210
C	PFU01220
C	PFU01230
C	PFU01240
C	PFU01250
C	PFU01260
C	PFU01270
C	PFU01280
C	PFU01290
C	PFU01300
C	PFU01310
C	PFU01320
C	PFU01330
C	PFU01340
C	PFU01350
C	PFU01360
C	PFU01370
C	PFU01380
C	PFU01390
C	PFU01400

```

DO 1060 I=1,NWVL
IF((WVL(I).LT.0.19).OR.(WVL(I).GT.12.049)) GO TO 504
IF(WVL(I).LE.0.8) WVL(I)=0.55
IF(WVL(I).GT.0.8).AND.(WVL(I).LE.2.1) WVL(I)=1.06
IF(WVL(I).GT.12.0) WVL(I)=12.0
IF(WVL(I).GT.5.0).AND.(WVL(I).LE.5.25) WVL(I)=5.0
IF(WVL(I).GE.7.6).AND.(WVL(I).LT.8.0) WVL(I)=8.0
IF(WVL(I).GE.2.85).AND.(WVL(I).LT.3.0) WVL(I)=3.0
IF(WVL(I).GT.2.1).AND.(WVL(I).LT.2.85) GO TO 498
IF(WVL(I).GT.5.25).AND.(WVL(I).LT.7.6) GO TO 498
1060 CONTINUE
1070 CONTINUE
C*** MAIN INTERPOLATION LOOP
C
DO 2000 I=1,NWVL
IF(I.GT.1) GO TO 1260

```

```

1100 CONTINUE
READ(IPHFUN,300) IANG,ID,WAVEH,ALBH,BEXH,BSCH
READ(IPHFUN,400) (PFH(L),L=1,LMAX)
IF((IDN.EQ.0).AND.(.NOT.ISPOT)) GO TO 1280
IF(WVL(I).LT.WAVEH) GO TO 1100
1150 CONTINUE
READ(IPHFUN,300) IANG,ID,WAVE,ALBE,BEX,BSC
READ(IPHFUN,400) (PF(L),L=1,LMAX)
IF(WVL(I).LE.WAVE) GO TO 1260
1160 CONTINUE
WAVEH=WAVE
ALBH=ALBE
BEXH=BEX
DO 1240 L=1,LMAX
1240 PFH(L)=PF(L)
GO TO 1150
1260 CONTINUE
C
C*** GO TO NEXT WAVELENGTH INTERPOLATION INTERVAL IF INPUT WAVE-
C*** LENGTH IS GREATER THAN THE MAXIMUM OF THE CURRENT ONE.
C
IF(WVL(I).GT.WAVE) GO TO 1160
C
C*** RENORMALIZE LOWER END OF INTERPOLATION INTERVAL
C
1280 CONTINUE
SUM(1)=0.0
DO 1200 L=2,LMAX
1200 SUM(L)=((ANG(L-1)-ANG(L))*(PFH(L-1)+PFH(L))/4.0)+SUM(L-1)
SUMT=SUM(LMAX)
DO 1250 L=1,LMAX
1250 PFH(L)=PFH(L)/SUMT
C
C*** BRANCH TO FINAL PROCEDURE FOR BASCAT USER-DEFINED PHASE
C*** FUNCTION IF APPROPRIATE.
C
IF((IDN.EQ.0).AND.(.NOT.ISPOT)) GO TO 2500
C
C*** RENORMALIZE UPPER END OF INTERPOLATION INTERVAL.
C
SUM(1)=0.0
DO 1400 L=2,LMAX
1400 SUM(L)=((ANG(L-1)-ANG(L))*(PF(L-1)+PF(L))/4.0)+SUM(L-1)
SUMT=SUM(LMAX)
DO 1450 L=1,LMAX
1450 PF(L)=PF(L)/SUMT
C
C*** BRANCH TO BASCAT WAVELENGTH INTERPOLATION PROCEDURE IF
C*** APPROPRIATE.
C
IF(.NOT.ISPOT) GO TO 2500
C
C*** PERFORM HALVING SEARCH FOR COSINES IN PHASE FUNCTION DATA FILE
C*** WHICH BRACKET COSINE INPUT FROM SPOT.
C
L=1
LL=LMAX-1
DO 1300 K=1,KMAX
LL=LL/2
L=L+LL
AT=COSIN-ANG(L)
IF(AT.GT.0.) L=L-LL
1300 CONTINUE
C
C*** PERFORM SPOT EXTINCTION COEFFICIENT AND PHASE FUNCTION
C*** INTERPOLATIONS OVER WAVELENGTH AND ANGLE.
C
FACANG=(COSIN-ANG(L))/(ANG(L+1)-ANG(L))
FACWVL=(WVL(I)-WAVEH)/(WAVE-WAVEH)
BE(I)=BEXH+(BEX-BEXH)*FACWVL

```

```

PFU01410
PFU01420
PFU01430
PFU01440
PFU01450
PFU01460
PFU01470
PFU01480
PFU01490
PFU01500
PFU01510
PFU01520
PFU01530
PFU01540
PFU01550
PFU01560
PFU01570
PFU01580
PFU01590
PFU01600
PFU01610
PFU01620
PFU01630
PFU01640
PFU01650
PFU01660
PFU01670
PFU01680
PFU01690
PFU01700
PFU01710
PFU01720
PFU01730
PFU01740
PFU01750
PFU01760
PFU01770
PFU01780
PFU01790
PFU01800
PFU01810
PFU01820
PFU01830
PFU01840
PFU01850
PFU01860
PFU01870
PFU01880
PFU01890
PFU01900
PFU01910
PFU01920
PFU01930
PFU01940
PFU01950
PFU01960
PFU01970
PFU01980
PFU01990
PFU02000
PFU02010
PFU02020
PFU02030
PFU02040
PFU02050
PFU02060
PFU02070
PFU02080
PFU02090
PFU02100

```

	PFSPOT(I)=PFH(L)*(1.-FACANG-FACWVL+FACANG+FACWVL)+	PFU02120
	+PFH(L+1)*(FACANG-FACANG+FACWVL)+PF(L)*(FACWVL-FACANG+FACWVL)+	PFU02130
	+PF(L+1)*(FACANG+FACWVL)	PFU02140
2000	CONTINUE	PFU02150
C		PFU02160
C***	LOAD FIRST NWVL ANGLES OF OUTPUT ARRAY PF() WITH INTERPOLATED	PFU02170
C***	RESULTS FOR SPOT.	PFU02180
C		PFU02190
	DO 2200 N=1,NWVL	PFU02200
2200	PF(N)=PFSPOT(N)/(4.*PI)	PFU02210
C		PFU02220
C***	FINAL EXIT FOR SPOT PROCESSING.	PFU02230
C		PFU02240
	GO TO 500	PFU02250
2500	CONTINUE	PFU02260
	FACWVL=0.	PFU02270
	IF(IDN.EQ.0) GO TO 2700	PFU02280
C		PFU02290
C***	BASCAT ALBEDO, EXTINCTION COEFFICIENT, AND PHASE FUNCTION	PFU02300
C***	INTERPOLATION OVER WAVELENGTH.	PFU02310
C		PFU02320
	FACWVL=(WVL(1)-WAVEH)/(WAVE-WAVEH)	PFU02330
2700	CONTINUE	PFU02340
	DO 2800 L=1,LMAX	PFU02350
2800	PF(L)=PFH(L)+(PF(L)-PFH(L))*FACWVL	PFU02360
	ALB(L)=ALBH+(ALBE-ALBH)*FACWVL	PFU02370
	BEX(L)=BEXH+(BEX-BEXH)*FACWVL	PFU02380
C		PFU02390
C***	FINAL EXIT FOR BASCAT USER-DEFINED PHASE FUNCTION PROCEDURE.	PFU02400
C		PFU02410
	IF(IDN.EQ.0) GO TO 500	PFU02420
C		PFU02430
C***	FINAL BASCAT PHASE FUNCTION RENORMALIZATION.	PFU02440
C		PFU02450
	SUM(1)=0.0	PFU02460
	DO 2900 L=2,LMAX	PFU02470
2900	SUM(L)=(ANG(L-1)-ANG(L))*(PF(L-1)+PF(L))/4.0+SUM(L-1)	PFU02480
	SUMT=SUM(LMAX)	PFU02490
	DO 2950 L=1,LMAX	PFU02500
2950	PF(L)=PF(L)/SUMT	PFU02510
C		PFU02520
C***	FINAL EXIT FOR BASCAT PROCESSING.	PFU02530
C		PFU02540
	GO TO 500	PFU02550
C		PFU02560
C***	ERROR EXIT BLOCK COMMON TO SPOT AND BASCAT	PFU02570
C		PFU02580
491	CONTINUE	PFU02590
	WRITE(IOOUT,495)	PFU02600
495	FORMAT(1H0,20X,58H***PFUNC ERROR*** AEROSOL ID NUMBER OUT OF ALLOW	PFU02610
	+ABLE RANGE //)	PFU02620
	STOP	PFU02630
492	CONTINUE	PFU02640
	WRITE(IOOUT,496)	PFU02650
496	FORMAT(1H0,20X,83H***PFUNC ERROR*** READ TERMINATION SENTINEL NOT	PFU02660
	+FOUND OR NUMBER OF ANGLES EXCEED 65 //)	PFU02670
	STOP	PFU02680
493	CONTINUE	PFU02690
	WRITE(IOOUT,497)	PFU02700
497	FORMAT(1H0,20X,93H***PFUNC ERROR*** NUMBER OF SPECIFIED ANGLES AND	PFU02710
	+NUMBER OF PHASE FUNCTION VALUES DO NOT MATCH //)	PFU02720
	STOP	PFU02730
498	CONTINUE	PFU02740
	WRITE(IOOUT,499)	PFU02750
499	FORMAT(1H0,10X,79H***PFUNC ERROR*** SOME OR ALL WAVELENGTHS IN WVP	PFU02760
	+L INPUT ARRAY DO NOT LIE WITHIN /1H,45HWAVELENGTH BANDS COVERED	PFU02770
	+Y PFENDAT DATA BASE //)	PFU02780
	STOP	PFU02790
504	CONTINUE	PFU02800
	WRITE(IOOUT,505)	PFU02810

505 FORMAT(1H0,10X,109H***PFUNC ERROR*** SOME OR ALL WAVELENGTHS IN WVPFU02820
+L ARRAY DO NOT LIE WITHIN OVERALL ACCEPTABLE RANGE OF 0.2-12.0 /> PFU02830
500 STOP RETURN PFU02840
END PFU02850
PFU02860

```

SUBROUTINE XSCALE(WAVE,VIS,EXT55,XSTRN,IERR,ISLT,IFOG,RANGE,ANGLE)XSC00010
THE PURPOSE OF THIS ROUTINE IS TO A) FIND THE HORIZONTAL EXTINCTIONXSC00020
IN FOG AT THE WAVELENGTHS SPECIFIED BELOW FROM THE EXTINCTIONXSC00030
AT .55 UM OR B) THE EXTINCTION ALONG A SLANT PATH AT ALLOWEDXSC00040
WAVELENGTHS FROM THE EXTINCTION AT .55 UM: FOG TYPE 1, 2, OR 3,XSC00050
MUST BE SPECIFIED FOR SLANT PATHS.XSC00060
*** VISIBILITY = 88. OR 89. IS NOT ALLOWED AS THIS IS USED AS ANXSC00070
INDICATOR THAT XSCALE IS BEING CALLED AS A SUBROUTINE FROM EITHERXSC00080
SPOT(88.), LOWTRAN(88.), OR CWIC(89.), NOT EOMAIN!XSC00090
WAVE=LAMDA IN UM - MUST BE .55, 1.06, 3.0-5.0, 8.0-12.05.XSC00100
*** ALL EXTN'S ARE IN KM**-1XSC00110
EXT55 = EXTINCTION AT .55 UMXSC00120
EXT106 = EXTINCTION AT 1.06 UMXSC00130
EXT35 = EXTINCTION FROM 3.0 TO 5.0 UMXSC00140
EXT812 = EXTINCTION FROM 8.0 TO 12.0 UMXSC00150
VIS= VISIBILITY IN KM -OR- EXT55 IN KM**-1XSC00160
EXT55 IS ** NOT ** CHANGED BY THIS ROUTINE.XSC00170
*****XSC00180
INPUT: THERE IS A MAXIMUM OF 3 CARDS TO EXECUTE THIS MODULEXSC00190
THE CARDS MAY BE INSERTED IN ANY ORDER WITH THE EXCEPTION OFXSC00200
THE LAST CARD WHICH SIGNIFIES THAT EXECUTION IS TO BEGIN.XSC00210
THE CARDS ARE INPUT WITH FORMAT (A4,6X,5(F6.2,1X)).XSC00220
EACH CARD BEGINS WITH A 4 LETTER IDENTIFIER IN COL 1 - 4XSC00230
FOLLOWED BY AS MANY (REAL) FIELDS AS NEEDED, 6 COL PERXSC00240
FIELD BEGINNING IN COL 11, WITH A BLANK BETWEEN EACH SUBSEQUENTXSC00250
FIELD. THE CARDS ARE NOT ORDER DEPENDENT.XSC00260
IF GEOMET OPTION IS BEING USED, THEN ONLY THE IDENTIFIER HORZ,XSC00270
SLNH, OR SLNS IS TO BE READ IN (NO ADDITIONAL PARAMETERS NEEDED).XSC00280
FOG FOG TYPE, RAIN RATE (MM/HR); RAIN RATE ONLY NEEDEDXSC00290
WHEN FOG TYPE=4,XSC00300
HORZ HORDIS (KM); HORIZONTAL PATH CALCULATIONXSC00310
SLNH HORDIS (KM), ANGLE (DEGREES); SLANT PATH CALCULATIONXSC00320
SLNS SLTDIS (KM), ANGLE (DEGREES); SLANT PATH CALCULATIONXSC00330
PLOT WRITE SLANT PATH EXTINCTION, AT INPUT WAVELENGTH, ANDXSC00340
ALTITUDE TO NPLTU (SEE COMMON BLOCK IOUNIT); THEXSC00350
FIRST RECORD WILL BE THE NUMBER OF POINTS TO BE WRITTEN,XSC00360
FORMATS: RECORD 1 (15), SUBSEQUENT RECORDS (2(E10.4,1X))
GO SIGNIFIES TO BEGIN EXECUTION, NO MORE INPUT FORXSC00370
THIS CALL. NOTE THAT IF A DATA CARD IS NOT READXSC00380
THEN ANY VALUES ESTABLISHED FROM PREVIOUS CALLSXSC00390
TO THE MODULE ARE STILL IN EFFECT.XSC00400
ALL THE FOLLOWING FOG TYPES ARE RELEVANT TO HORIZONTAL PATHS,XSC00410
BUT ONLY FOG TYPES 1, 2, OR 3 ARE ALLOWED FOR SLANT PATH CALCULATIONSXSC00420
FOG TYPE=1. FOR MARITIME ARTICXSC00430
=2. FOR MARITIME POLARXSC00440
=3. FOR CONTINENTAL POLARXSC00450
=4. FOR RAINXSC00460
=5. FOR SNOWXSC00470
HORDIS - HORIZONTAL DISTANCE IN KM.XSC00480
SLTDIS - SLANT PATH DISTANCE IN KM.XSC00490
ANG= LOOK ANGLE FROM HORIZONTAL IN DEGREESXSC00500
N.B. ONE OF THE FOLLOWING COMBINATIONS MUST BE SUPPLIEDXSC00510
FOR SLANT PATH CALCULATIONS.XSC00520
HORDIS AND ANG ** OR ** SLTDIS AND ANGXSC00530
*****XSC00540
OUTPUTXSC00550
TRANSMISSION AT APPROPRIATE WAVELENGTH FOR SLANT OR HORIZONTAL PATHXSC00560
COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGKXSC00570
COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUXSC00580
COMMON /GEOMET/PTS(15),IGEOSWXSC00590
DIMENSION TYPE(6),DAT(6)XSC00600
LOGICAL NOLOXSC00610
DATA A0,A1,A2,NPLT/0.1425,0.1475,-0.0017,0/XSC00620
USE VIS=88., OR 89. AS AN INDICATOR THAT XSCALE HAS BEEN CALLEDXSC00630
AS A SUBROUTINE FROM OTHER PROGRAMS - NOT EOMAIN!XSC00640
IF (VIS.LT.87.9.OR.VIS.GT.89.1) GO TO 8XSC00650
ANG=ANGLEXSC00660
FIND ELEVATION ANGLE FROM ZENITH ANGLE IN SPOT AND LOWTRANXSC00670

```

	IF (VIS.GT.87.9.AND.VIS.LT.88.1) ANG=90.-ANGLE	XSC00650
	IF (VIS.GT.87.9.AND.VIS.LT.88.1.AND.ANGLE.GT.90.) ANG=ANGLE-90.	XSC00660
	HORDIS=0.	XSC00670
	SLTDIS=0.	XSC00680
	NOLQ=.FALSE.	XSC00690
C	ISLT=0=HORIZONTAL; ISLT.GT.0 = SLANT.	XSC00700
	IF (ISLT.EQ.0) HORDIS=RANGE	XSC00710
	IF (ISLT.EQ.0) ISLANT=0	XSC00720
	IF (ISLT.GT.0) SLTDIS=RANGE	XSC00730
	IF (ISLT.GT.0) ISLANT=1	XSC00740
	GO TO 6	XSC00750
8	CONTINUE	XSC00760
	NOLQ=.TRUE.	XSC00770
	DO 9 I=1,3	XSC00780
	READ (IOIN,500) (DAT(J),J=1,5)	XSC00790
	IF (DAT(1).EQ.TYPE(1)) GO TO 1	XSC00800
	IF (DAT(1).EQ.TYPE(2)) GO TO 2	XSC00810
	IF (DAT(1).EQ.TYPE(3)) GO TO 3	XSC00820
	IF (DAT(1).EQ.TYPE(4)) GO TO 4	XSC00830
	IF (DAT(1).EQ.TYPE(5)) GO TO 5	
	IF (DAT(1).EQ.TYPE(6)) GO TO 6	XSC00840
C	ERROR CHECK	XSC00850
	GO TO 7	XSC00860
C	ADVERSE WEATHER INDICATOR AND OPTIONAL RAIN RATE:	XSC00870
1	IFOG=IFIX(DAT(2))	XSC00880
	RHRT=DAT(3)	XSC00890
	GO TO 9	XSC00900
C	HORIZONTAL DISTANCE FOR HORIZONTAL PATH CALC.	XSC00910
2	HORDIS=DAT(2)	XSC00920
	ISLANT=0	XSC00930
	GO TO 9	XSC00940
C	HORIZONTAL DISTANCE AND ANGLE FOR SLANT PATH CALC.	XSC00950
3	HORDIS=DAT(2)	XSC00960
	ANG=DAT(3)	XSC00970
	ISLANT=1	XSC00980
	GO TO 9	XSC00990
C	SLANT DISTANCE AND ANGLE FOR SLANT PATH CALC.	XSC01000
4	SLTDIS=DAT(2)	XSC01010
	ANG=DAT(3)	XSC01020
	ISLANT=1	XSC01030
	GO TO 9	
C	SET PLOT FLAG	
5	NPLT=1	
9	CONTINUE	XSC01040
6	CONTINUE	XSC01050
	IF (NOLQ) WRITE (IOOUT,600)	XSC01060
	IF (IGEOSW.NE.1) GO TO 88	XSC01070
	HORDIS=SQRT((PTS(1)-PTS(4))*2+(PTS(2)-PTS(5))*2)	XSC01080
	SLTDIS=SQRT(HORDIS**2+(PTS(3)-PTS(6))*2)	XSC01090
	ANG=ACOS(HORDIS/SLTDIS)/PIRAD	XSC01100
88	CONTINUE	XSC01110
C	WAVELENGTH ERROR CHECK	XSC01120
	IF ((WAVE.GT..4.AND.WAVE.LE.2.) .OR. (WAVE.GE.3..AND.WAVE.LE.5.)	XSC01130
1	.OR. (WAVE.GE.8.AND.WAVE.LE.12.05)) GO TO 10	XSC01140
	WRITE (IOOUT,1600) WAVE	XSC01150
	IERR=1	XSC01160
	XSTRN=1.	XSC01170
	RETURN	XSC01180
10	CONTINUE	XSC01190
	IF (NOLQ.AND.IFOG.EQ.1) WRITE (IOOUT,800)	XSC01200
	IF (NOLQ.AND.IFOG.EQ.2) WRITE (IOOUT,900)	XSC01210
	IF (NOLQ.AND.IFOG.EQ.3) WRITE (IOOUT,950)	XSC01220
	IF (NOLQ.AND.IFOG.EQ.4) WRITE (IOOUT,1000)	XSC01230
	IF (NOLQ.AND.IFOG.EQ.5) WRITE (IOOUT,1100)	XSC01240
	IF (ISLANT.GT.0) GO TO 11	XSC01250
	IF (NOLQ) WRITE (IOOUT,1200)	XSC01260
11	IF (NOLQ.AND.ISLANT.GE.1) WRITE (IOOUT,1400) WAVE	XSC01270
	IF (ISLANT.GE.1.AND.(IFOG.LE.0.OR.IFOG.GE.4)) WRITE (IOOUT,2100)	
	IF (ISLANT.GE.1.AND.(IFOG.LE.0.OR.IFOG.GE.4)) IFOG=1	
	EXTN=EXT55	

```

12 IF (ISLANT.GE.1.AND.WAVE.GE.8) GO TO 101
   IF (ISLANT.GE.1) CALL SLANT<EXTN,HORDIS,SLTDIS,ANG,AVEX55,IERR,
1 WAVE,NPLT>
   IF (IERR.EQ.1) XSTRN = 1.
   IF (IERR.EQ.1) RETURN
   EXTN=EXT55
   IF (ISLANT.GE.1) EXTN=AVEX55
   XSPATH=HORDIS
   IF (ISLANT.GE.1) XSPATH=SLTDIS
   IF (ISLANT.EQ.0) GO TO 100
C SLANT PATH EXTINCTION
  XSTRN =EXP<-XSPATH*EXTN>
  IF (NOLO) WRITE (IOOUT,1500) WAVE,EXTN,XSPATH,XSTRN,ANG
  RETURN
100 IF (IFOG.NE.4) GO TO 101
C RAIN - ALL WAVELENGTHS
  IF (NOLO.AND.RNRT.LE.0.0) WRITE (IOOUT,550)
  IF (RNRT.LE.0.0) RNRT=1.
  RNEXTN=A0+A1*RNRT+A2*RNRT**2
  XSTRN =EXP<-XSPATH*RNEXTN>
  IF (NOLO) WRITE (IOOUT,1550) RNRT,RNEXTN,XSPATH,XSTRN
  RETURN
101 IF (ABS(WAVE-1.06).LT.01) GO TO 400
   IF (WAVE.GE.3.AND.WAVE.LE.5) GO TO 200
   IF (WAVE.GE.8.AND.WAVE.LE.12) GO TO 300
   XSTRN=EXP<-XSPATH*EXTN>
   RETURN
200 CONTINUE
C 3.0 TO 5.0 RANGE
C MA
C IF (IFOG.EQ.1) EXT35=10.**(<math>0.0345+1.03*\log_{10}(EXTN)</math>)>
C MP
C IF (IFOG.EQ.2) EXT35=10.**(<math>-0.38+1.32*\log_{10}(EXTN)</math>)>
C CP
C IF (IFOG.EQ.3) EXT35=10.**(<math>-0.82+1.58*\log_{10}(EXTN)</math>)>
C IF (IFOG.NE.5) EXTN=EXT35
C IF (ISLANT.GE.1) GO TO 12
C SNOW
C IF (IFOG.EQ.5) EXT35=10.0**(<math>1.05*\log_{10}(EXTN)+.021</math>)>
C XSTRN =EXP<-XSPATH*EXT35>
C IF (NOLO) WRITE (IOOUT,1700) EXT35,XSPATH,XSTRN
C RETURN
300 CONTINUE
C 8.0 TO 12.0 RANGE
C MA
C IF (IFOG.EQ.1) EXT812=10.**(<math>-0.45+1.19*\log_{10}(EXTN)</math>)>
C MP
C IF (IFOG.EQ.2) EXT812=10.**(<math>-1.01+1.51*\log_{10}(EXTN)</math>)>
C CP
C IF (IFOG.EQ.3) EXT812=10.**(<math>-1.65+1.82*\log_{10}(EXTN)</math>)>
C IF (IFOG.NE.5) EXTN=EXT812
C IF (ISLANT.GE.1) GO TO 12
C SNOW
C IF (IFOG.EQ.5) EXT812=10.0**(<math>.993*\log_{10}(EXTN)+.114</math>)>
C XSTRN =EXP<-XSPATH*EXT812>
C IF (NOLO) WRITE (IOOUT,1800) EXT812,XSPATH,XSTRN
C RETURN
400 CONTINUE
C 1.06 RANGE
C MA, MP, AND CP
C EXT106=AMIN1<math>(10.**(-0.14+1.16*\log_{10}(EXTN)),EXTN)</math>>
C EXTN=EXT106
C IF (ISLANT.GE.1) GO TO 12
C SNOW - ASSUME THAT THE EXTINCTION AT 1.06 IS THE SAME AS AT .55 UMX
C IF (IFOG.EQ.5) EXT106=EXT55
C XSTRN =EXP<-XSPATH*EXT106>
C IF (NOLO) WRITE (IOOUT,1900) EXT106,XSPATH,XSTRN
C RETURN
7 WRITE (IOOUT,2000) <math>(DAT(J),J=1,4)</math>
  XSTRN=1.

```

```

XSC01280
XSC01290
XSC01300
XSC01310
XSC01320
XSC01330
XSC01340
XSC01350
XSC01360
XSC01370
XSC01380
XSC01390
XSC01400
XSC01410
XSC01420
XSC01430
XSC01440
XSC01450
XSC01460
XSC01470
XSC01480
XSC01490
XSC01500
XSC01510
XSC01520
XSC01530
XSC01540
XSC01550
XSC01560
XSC01570
XSC01580
XSC01590
XSC01600
XSC01610
XSC01620
XSC01630
XSC01640
XSC01650
XSC01660
XSC01670
XSC01680
XSC01690
XSC01700
XSC01710
XSC01720
XSC01730
XSC01740
XSC01750
XSC01760
XSC01770
XSC01780
XSC01790
XSC01800
XSC01810
XSC01870
XSC01880
XSC01890
XSC01900
XSC01910
XSC01920
XSC01930

```

	IERR=1		XSC01940
	RETURN		XSC01950
500	FORMAT (A4,6X,5(F6.2,1X))		XSC01960
550	FORMAT(1H0,50HXSCALE WARNING - RAIN RATE IS .LE. 0.0, RAIN RATE ,		XSC01970
	+20HSET EQUAL TO 1 MM/HR,/))		XSC01980
600	FORMAT (1H,///,1X,50X,14HOPTIONS CHOSEN)		XSC01990
800	FORMAT (1H,50X,14HMARITIME ARTIC)		XSC02000
900	FORMAT (1H,50X,14HMARITIME POLAR)		XSC02010
950	FORMAT (1H,50X,17HCONTINENTAL POLAR)		XSC02020
1000	FORMAT (1H,50X,4HRAIN)		XSC02030
1100	FORMAT (1H,50X,4HSNOW)		XSC02040
1200	FORMAT (1H,50X,15HHORIZONTAL PATH/)		XSC02050
1400	FORMAT (1H,50X,15HSLANT PATH FOR ,F8.3,8H MICRONS/)		XSC02060
1500	FORMAT (1X,23X,20HSLANT EXTINCTION AT ,F5.2,8H MICRONS,		XSC02070
	4X,11HDISTANCE ,12HTRANSMISSION,6X,5HANGLE/,1X,40X,		XSC02080
1	26HKM**-1 KM, /		XSC02090
2	1X,38X,F8.3,12X,F8.3,5X,E9.3,F7.2)		XSC02100
3	1550 FORMAT (1H,20X,27HEXTINCTION FOR RAIN RATE OF,F5.2,6H MM/HR,		XSC02110
1	5X,11HDISTANCE ,12HTRANSMISSION/,1X,40X,		XSC02120
2	26HKM**-1 KM, /		XSC02130
3	1X,38X,F8.3,12X,F8.3,5X,E9.3)		XSC02140
1600	FORMAT (1H,18H***** WAVELENGTH (,F9.3,10H) OUTSIDE ,		XSC02150
1	10HALLOWABLE ,29HRANGE (1.06,3.0-5.0,8.0-12.0		XSC02160
2	31HMICRONS) - CONTROL RETURNED TO ,		XSC02170
3	17HMAIN FROM XSCALE.)		XSC02180
1700	FORMAT (1H,25X,37HEXTINCTION FROM 3.0 TO 5.0 MICRONS		XSC02190
1	11HDISTANCE ,12HTRANSMISSION/,1X,40X,		XSC02200
2	26HKM**-1 KM, /		XSC02210
3	1X,38X,F8.3,13X,F8.3,5X,E9.3)		XSC02220
1800	FORMAT (1H,25X,37HEXTINCTION FROM 8.0 TO 12.0 MICRONS		XSC02230
1	12H DISTANCE ,12HTRANSMISSION/,1X,40X,		XSC02240
2	26HKM**-1 KM, /		XSC02250
3	1X,38X,F8.3,12X,F8.3,5X,E9.3)		XSC02260
1900	FORMAT (1H,25X,37HEXTINCTION AT 1.06 MICRONS		XSC02270
1	11HDISTANCE ,12HTRANSMISSION/,1X,40X,		XSC02280
2	26HKM**-1 KM, /		XSC02290
3	1X,38X,F8.3,12X,F8.3,5X,E9.3)		XSC02300
2000	FORMAT (1H,44HUNKNOWN CARD TYPE. CONTROL RETURNED TO MAIN,		XSC02310
	+13H FROM XSCALE,/,1X,A4,6X,5(F6.2,1X))		XSC02320
2100	FORMAT(1H,40HINCORRECT FOG TYPE FOR SUBROUTINE SLANT,/,1X,		XSC02330
	1 21HFOG TYPE CHANGED TO 1/)		
	END		XSC02340

```

SUBROUTINE SLANT(EXT55,HORDIS,SLTDIS,ANG,AVEX55,IERR,WAVE,NPLT) SLNT0010
COMMON /IOUNIT,IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUSLNT0020
COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK SLNT0030
C ALL QUANTITIES IN THIS ROUTINE ARE FOR .55 UM SLNT0040
REAL KMTOM SLNT0050
C KILOMETERS TO METERS SLNT0060
KMTOM=1000. SLNT0070
IERR=0 SLNT0080
TOL=.0001 SLNT0090
100 IF (HORDIS,LT,TOL,OR,ANG,LT,TOL) GO TO 200 SLNT0100
C HORIZONTAL DISTANCE AND ANGLE INPUT SLNT0110
VERDIS=HORDIS*TAN(ANG*PIRAD) SLNT0120
SLTDIS=SQRT(HORDIS**2+VERDIS**2) SLNT0130
GO TO 300 SLNT0140
200 IF (SLTDIS,LT,TOL,OR,ANG,LT,TOL) GO TO 500 SLNT0150
C SLANT DISTANCE AND ANGLE INPUT SLNT0160
VERDIS=SLTDIS*SIN(ANG*PIRAD) SLNT0170
HORDIS=SQRT(SLTDIS**2-VERDIS**2) SLNT0180
C CONVERT TO 20 METER INCREMENTS SLNT0190
300 VERDIS=FLOAT(IFIX((VERDIS+TOL)*KMTOM/20.))*20./KMTOM SLNT0200
C LIMIT ON VERTICAL HEIGHT IS 500 METERS SLNT0210
IF (VERDIS*KMTOM,GT,500.) VERDIS=.5 SLNT0220
SD=SQRT(HORDIS**2+VERDIS**2) SLNT0230
IF (SD/SLTDIS,GT,1.01,OR,SD/SLTDIS,LT,.99)WRITE(IOOUT,700)SLTDIS,SD SLNT0240
SLTDIS=SD SLNT0250
C FIND NBR OF 20 METER INCREMENTS SLNT0260
ITOP=IFIX((VERDIS+TOL)*KMTOM/20.) SLNT0270
IF (ITOP,LT,1) ITOP=1
VERDIS=FLOAT(ITOP)*20./KMTOM
EXTN=EXT55 SLNT0280
C BEGIN TRAPEZODIAL INTEGRATION FOR TAU ( OPTICAL DEPTH) SLNT0290
TAU=EXTN/2. SLNT0300
ALT=0.0
NPTS=ITOP+1
IF (NPLT,EQ,1) WRITE (NPLOTU,352) NPTS
352 FORMAT (I5)
IF (NPLT,EQ,1) WRITE (NPLOTU,351) EXTN,ALT,WAVE
DO 400 I=1,ITOP SLNT0310
C THESE FORMULAS ARE GOOD ONLY IN 20M INCREMENTS SLNT0320
IF (EXTN,GE,7.0,AND,WAVE,LT,2.0) EXTN= SLNT0330
1 10.**(0.55+0.72*ALOG10(EXTN)) SLNT0340
IF (EXTN,LT,7.0,AND,WAVE,LT,2.0) EXTN= SLNT0350
1 10.**(0.1+1.25*ALOG10(EXTN)) SLNT0360
IF (EXTN,GE,3.3,AND,(WAVE,GE,3.0,AND,WAVE,LT,5.0)) EXTN= SLNT0370
1 10.**(0.55+.72*ALOG10(EXTN)) SLNT0380
IF (EXTN,LT,3.3,AND,(WAVE,GE,3.0,AND,WAVE,LT,5.0)) EXTN= SLNT0390
1 10.**(0.3+1.2*ALOG10(EXTN)) SLNT0400
IF (EXTN,GE,1.7,AND,(WAVE,GE,8.0,AND,WAVE,LT,12.0)) EXTN= SLNT0410
1 10.**(0.5+0.75*ALOG10(EXTN)) SLNT0420
IF (EXTN,LT,1.7,AND,(WAVE,GE,8.0,AND,WAVE,LT,12.0)) EXTN= SLNT0430
1 10.**(0.4+1.2*ALOG10(EXTN)) SLNT0440
ALT=FLOAT(I)*20.
IF (NPLT,EQ,1) WRITE (NPLOTU,351) EXTN,ALT
351 FORMAT (3E10.4,1X))
400 TAU=TAU+EXTN SLNT0450
C FINISH TRAP INTEGRATION SLNT0460
TAU=(TAU-EXTN/2.)*.02 SLNT0470
C FIND AVERAGE EXTINCTION VALUE FOR SLANT PATH. SLNT0480
AVEX55=TAU/VERDIS SLNT0490
RETURN SLNT0500
500 WRITE (IOOUT,600) SLNT0510
IERR=1 SLNT0520
RETURN SLNT0530
C SLNT0540
600 FORMAT (1X,38HERROR - IMPROPER INPUT FOR SUBROUTINE SLNT0550
1 34HSLANT, TRANSMISSION SET EQUAL TO 1) SLNT0560
700 FORMAT (1H,18HWARNING FROM SLANT,/,1X,22HTHE VERTICAL DISTANCE , SLNT0570
+38HEXCEEDS THE 500 METER UPPER LIMIT, OR ,/,1X,10HIS NOT AN SLNT0580
+29HINTEGER MULTIPLE OF 20 METERS,/,1X,28HSLANT DISTANCE CHANGED FR SLNT0590
+0M ,F7.4,4H TO ,F7.4,3H KM/) SLNT0600

```

END

SLNT0610

SUBROUTINE TURB(WAVE, IERR)
CALCULATES TURBULENCE INDUCED POINTING JITTER AND POWER SPECTRUM
FOR LASER TARGET DESIGNATOR AND TERMINAL HOMING SEEKER

CALCULATION FOR THE DESIGNATOR PATH ARE PERFORMED EACH
TIME THIS ROUTINE IS REFERENCED. THE CALCULATIONS FOR THE
SEEKER PATHS ARE PERFORMED ONLY WHEN THE DATA CARDS DVRV,
CN2, OR V2 ARE INCLUDED IN THE INPUT SET.
THE INPUT IS CARD ORDER INDEPENDENT, WITH THE SINGLE
RESTRICTION THAT THE 'GO' CARD MUST BE THE LAST CARD
OF THE DATA SET.
THE DATA IS COMPLETELY IDENTIFIED BY THE ID IN COLUMNS
1-4 OF EACH CARD, FOLLOWED BY UP TO 7<REAL> FIELDS AS
NEEDED, WITH 10 COLUMNS PER FIELD BEGINNING IN COL. 11.
COMMENTS BELOW.

THE INPUT FORMAT IS A4,6X,7(E10,4)

THE INPUT OF A CN1, CN2, V1 AND V2 TYPE DATA CARD IS
TERMINATED WHEN THE FIRST COEFFICIENT OF VALUE ZERO
IS ENCOUNTERED. THE REMAINING DATA COEFFICIENTS ON
THE CARD, IF ANY, ARE IGNORED.

THE FOLLOWING ARE REQUIRED RECORDS FOR AT THE FIRST INPUT SET.

IDENT. VARS. DESCRIPTION

PARM

DIAM LASER TARGET DESIGNATOR APERTURE DIAMETER IN METERS
THEY LASER BEAMSPREAD ANGLE IN RADIANS
TDOT LASER BEAM SLUE RATE IN RADIANS/SECOND
RANG DISTANCE FROM DESIGNATOR TO TARGET IN METERS
TIME DURATION OF CALCULATIONS IN SECONDS
M NO. OF FREQUENCIES FOR WHICH POINTING JITTER POWER
SPECTRUM IS TO BE CALCULATED, IF M=0, THEN DEFAULT
TO M=512.

***RANG AND R1V ARE RECALCULATED IF IGEOSW=1

***N1 AND N2 ARE CALCULATED WITHIN THE ROUTINE.

N1 NO OF SEGMENTS IN DESIGNATOR PATH
N2 NO. OF SEGMENTS IN SEEKER TO TARGET PATH

N1 AND N2 ARE SET EQUAL TO THE INDEX OF THE LAST NON-ZERO
COEFFICIENT READ INTO CN1 AND CN2 RESPECTIVELY.

CN1

IR2 STARTING INDEX VALUE OF CN1(I)
(CN1(I), I=IR2,IR2+5)
CN1(I) VALUES OF REFRACTIVE INDEX STRUCTURE CONSTANT
WITH ONE VALUE FOR EACH SEGMENT OF RANGE FROM LASER
DESIGNATOR TO TARGET (METERS**(-2/3))

V1

IR2 STARTING INDEX VALUE OF V1(I)
(V1(I), I=IR2,IR2+5)
V1(I) SET OF VALUES OF CROSSWIND VELOCITY
CORRESPONDING TO EACH SEGMENT OF RANGE FROM LASER
DESIGNATOR TO TARGET (M/SEC)

DVRV

D1V DIAMETER OF SEEKER APERTURE IN METERS.
R1V SEEKER RANGE TO TARGET IN METERS.

CN2

IR2 STARTING INDEX VALUE OF CN2(I)

TUP00010
TUR00020
TUR00030
TUR00040
TUR00050
TUR00060
TUR00070
TUR00080
TUR00090
TUR00100
TUR00110
TUR00120
TUR00130
TUR00140
TUR00150
TUR00160
TUR00170
TUR00180
TUR00190
TUR00200
TUR00210
TUR00220
TUR00230
TUR00240
TUR00250
TUR00260
TUR00270
TUR00280
TUR00290
TUR00300
TUR00310
TUR00320
TUR00330
TUR00340
TUR00350
TUR00360
TUR00370
TUR00380
TUR00390
TUR00400
TUR00410
TUR00420
TUR00430
TUR00440
TUR00450
TUR00460
TUR00470
TUR00480
TUR00490
TUR00500
TUR00510
TUR00520
TUR00530
TUR00540
TUR00550
TUR00560
TUR00570
TUR00580
TUR00590
TUR00600
TUR00610
TUR00620
TUR00630
TUR00640
TUR00650
TUR00660
TUR00670
TUR00680
TUR00690
TUR00700


```

C      (CN2(I), I=IR2,IR2+5)
C      CN2(I)  VALUES OF REFRACTIVE INDEX STRUCTURE
                CONSTANT FOR EACH SEGMENT OF RANGE FROM TARGET TO
                SEEKER (METERS**(-2/3))
V2      IR2      STARTING INDEX VALUE OF V2(I)
                V2(I)  VALUES OF CROSSWIND VELOCITY FOR EACH SEGMENT
                OF RANGE FROM TARGET TO SEEKER (M/SEC).
NPPS     NONE      PRINTS TABULAR VALUES OF POWER SPECTRUM VS FREQUENCY
NPPJ     NONE      PRINTS TABULAR VALUES OF POINTING JITTER VS TIME
NPAL     NONE      PRINTS TABULAR VALUES OF BOTH POWER SPECTRUM (PS)
                AND POINTING JITTER (PJ).
**IF THIS CARD IS MISSING NO TABLES WILL BE PRINTED (DEFAULT VALUE).
THE FOLLOWING IDENT RECORD IS ALWAYS REQUIRED.
GO      SIGNIFIES TO BEGIN EXECUTION FOR THIS DATA SET.
                AFTER EXECUTION, ANOTHER SET OF INPUTS MAY BE
                READ-IN FOLLOWED BY ANOTHER 'GO' CARD.
                ANY VALUES ESTABLISHED FROM PREVIOUS INPUT SETS
                TO THE ROUTINE ARE STILL IN EFFECT. THUS DATA
                SUCH AS FROM CARD PARM NEED NOT BE READ AGAIN IF
                THERE ARE TO BE NO CHANGES IN THE DATA ASSOCIATED
                WITH THAT IDENTIFIER.
*****
++ CALLED PROGRAMS ++
DESUB
FALPH
FFT4
GAUSS
MEANVR
RAND
SPECT
SPREAD
THETO
*****
C      COMPLEX RAN
      REAL LAMB
      REAL INR(7),IR1,LABEL(11)
      LOGICAL SETUP
      COMMON /CONST/PI,PI2,PIRAD,TWOP1,TORRMB,CDEGK
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
      THIS IS A COMPLEX NUMBER EACH NUMBER TAKES TWO WORDS
      COMMON /MOS/RAN(2048)
      COMMON /M01/FR(1025),CN1(20),V1(20),FO(20),RO(20)
      COMMON /LOWEX/PS(1025),V2(20),RR(10)
      DIMENSION CN2(20),PJCHAR(4),PSCHAR(4)
      COMMON /GEOMET/PTS(15),IGEOSW
      EXTERNAL DESUB,FALPH
      DATA SETUP/.TRUE./
      DATA PJCHAR,PSCHAR /4HPJ(,4HRAD*,4H*2/S,4HEC),4HPS(,4HRAD*,
1      4H*2/H,4HZ)/
      DATA LABEL/4HGO,4H,4HPARM,4HCN1,4HV1,4HDVVRV,4HCN2,
1      4HV2,4HNPPS,4HNPPJ,4HNPAL/
C      NPRINT=0
      LAMB=WAVE
C      SET THE SEED FOR THE RANDOM NUMBER GENERATOR
C*** NOTE, THIS SEED IS APPROPRIATE FOR THE RANDOM NUMBER GENERATOR
C*** USED AT THE ATMOSPHERIC SCIENCES LAB. USERS AT OTHER
C*** INSTALLATIONS WILL NEED TO SUPPLY THEIR OWN RANDOM NUMBER
C*** GENERATOR.
      IF (.NOT.SETUP) GO TO 100
      VARX=735.34829
      VARX=RAND(VARX)
      SETUP=.FALSE.

```

```

TUR00710
TUR00720
TUR00730
TUR00740
TUR00750
TUR00760
TUR00770
TUR00780
TUR00790
TUR00800
TUR00810
TUR00820
TUR00830
TUR00840
TUR00850
TUR00860
TUR00870
TUR00880
TUR00890
TUR00900
TUR00910
TUR00920
TUR00930
TUR00940
TUR00950
TUR00960
TUR00970
TUR00980
TUR00990
TUR01000
TUR01010
TUR01020
TUR01030
TUR01040
TUR01050
TUR01060
TUR01070
TUR01080
TUR01090
TUR01100
TUR01110
TUR01120
TUR01140
TUR01150
TUR01160
TUR01170
TUR01180
TUR01190
TUR01200
TUR01210
TUR01220
TUR01230
TUR01240
TUR01250
TUR01260
TUR01270
TUR01280
TUR01290
TUR01300
TUR01310
TUR01320
TUR01330
TUR01340
TUR01350
TUR01360
TUR01370
TUR01380
TUR01390
TUR01400

```

IOPT=1	TUR01410
100 READ(10IN,3000) IR1,(INR(I), I=1,7)	TUR01420
IR2=IFIX(INR(1))	TUR01430
IF<IR1.EQ.LABEL(1)> GO TO 180	TUR01440
IF<IR1.EQ.LABEL(3)> GO TO 110	TUR01450
IF<IR1.EQ.LABEL(4)> GO TO 120	TUR01460
IF<IR1.EQ.LABEL(5)> GO TO 130	TUR01470
IF<IR1.EQ.LABEL(6)> GO TO 140	TUR01480
IF<IR1.EQ.LABEL(7)> GO TO 150	TUR01490
IF<IR1.EQ.LABEL(8)> GO TO 160	TUR01500
C PRINTING OPTIONS	TUR01510
IF<IR1.EQ.LABEL(9)> NPRINT=1	TUR01520
IF<IR1.EQ.LABEL(10)> NPRINT=2	TUR01530
IF<IR1.EQ.LABEL(11)> NPRINT=3	TUR01540
IF(NPRINT.EQ.1.OR.NPRINT.EQ.2.OR.NPRINT.EQ.3) GO TO 100	TUR01550
WRITE(10OUT,3001) IR1,(INR(I), I=1,7)	TUR01560
GOTO 100	TUR01570
C	TUR01580
110 DIAM = INR(1)	TUR01590
THET = INR(2)	TUR01600
TDOT = INR(3)	TUR01610
RANG = INR(4)	TUR01620
TIME = INR(5)	TUR01630
M = IFIX(INR(6))	TUR01640
IF(M.EQ.0) M = 512	TUR01650
GOTO 100	TUR01660
C	TUR01670
120 DO 125 I=2,7	TUR01680
IF<INR(I).NE.0.0> GOTO 121	TUR01690
N1=IR2-1	TUR01700
GOTO 100	TUR01710
121 CN1<IR2>=INR(I)	TUR01720
IR2=IR2+1	TUR01730
IF<IR2.GT. 20> GOTO 126	TUR01740
125 CONTINUE	TUR01750
126 IR1=IR2-1	TUR01760
N1=MAX0(N1,IFIX(IR1))	TUR01770
GOTO 100	TUR01780
C	TUR01790
130 DO 131 I=2,7	TUR01800
V1<IR2>=INR(I)	TUR01810
IR2=IR2+1	TUR01820
IF<IR2.GT. 20> GOTO 100	TUR01830
131 CONTINUE	TUR01840
GOTO 100	TUR01850
C	TUR01860
140 IOPT = 2	TUR01870
O1V = INR(1)	TUR01880
R1V = INR(2)	TUR01890
GOTO 100	TUR01900
C	TUR01910
150 IOPT = 2	TUR01920
DO 155 I=2,7	TUR01930
IF<INR(I).NE.0.0> GOTO 151	TUR01940
N2=IR2-1	TUR01950
GOTO 100	TUR01960
151 CN2<IR2> = INR(I)	TUR01970
IR2=IR2+1	TUR01980
IF<IR2.GT. 20> GOTO 156	TUR01990
155 CONTINUE	TUR02000
156 IR1=IR2-1	TUR02010
N2=MAX0(N2,IFIX(IR1))	TUR02020
GOTO 100	TUR02030
C	TUR02040
160 IOPT = 2	TUR02050
DO 161 I=2,7	TUR02060
V2<IR2> = INR(I)	TUR02070
IR2=IR2+1	TUR02080
IF<IR2.GT. 20> GOTO 100	TUR02090
161 CONTINUE	TUR02100

	GOTO 100	TUR02110
C	180 IF(IGEOSW.NE.1)GO TO 190	TUR02120
	RANG=SQRT((PTS(7)-PTS(1))**2+(PTS(8)-PTS(2))**2+	TUR02130
	+ (PTS(9)-PTS(3))**2)	TUR02140
	DISKTM=1000.0	TUR02150
	RANG=RANG*DISKTM	TUR02160
	R1V=SQRT((PTS(4)-PTS(1))**2+(PTS(5)-PTS(2))**2+(PTS(6)-PTS(3))**2)	TUR02170
	R1V=R1V*DISKTM	TUR02180
	190 CONTINUE	TUR02190
	IF (LAMB.LE.14.) GO TO 200	TUR02200
	WRITE (100OUT,2500) LAMB	TUR02210
	IERR=1	TUR02220
	RETURN	TUR02230
	200 WRITE (100OUT,3100)	TUR02240
	IF (IOPT.EQ.2) WRITE (100OUT,3200)	TUR02250
	WRITE (100OUT,3300) LAMB,DIAM,THET	TUR02260
C	CHANGE WAVELENGTH TO METERS	TUR02270
	LAMB=LAMB/1.0E+6	TUR02280
	IF (IOPT.EQ.2) WRITE (100OUT,3400) D1V,R1V	TUR02290
	WRITE (100OUT,3500) TDOT,RANG	TUR02300
	WRITE (100OUT,3600) TIME	TUR02310
	WRITE (100OUT,3700) N1	TUR02320
	IF (IOPT.EQ.2) WRITE (100OUT,3800) N2	TUR02330
	WRITE (100OUT,3900) M	TUR02340
	WRITE (100OUT,4000)	TUR02350
	WRITE (100OUT,4200)	TUR02360
	DO 400 I=1,N1	TUR02370
	WRITE (100OUT,4300) I,CN1(I),V1(I)	TUR02380
	400 CONTINUE	TUR02390
	IF (IOPT.EQ.1) GO TO 600	TUR02400
	WRITE (100OUT,4100)	TUR02410
	WRITE (100OUT,4200)	TUR02420
	DO 500 I=1,N2	TUR02430
	WRITE (100OUT,4300) I,CN2(I),V2(I)	TUR02440
	500 CONTINUE	TUR02450
C	COMPUTATION OF TIME, FREQUENCY AND SPATIAL INCREMENTS	TUR02460
	600 DELT=TIME/M	TUR02470
	DELF=1./TIME	TUR02480
	DELZ=RANG/FLOAT(N1)	TUR02490
	IF (N2.NE.0) DEL1V=R1V/FLOAT(N2)	TUR02500
	MM=M+M	TUR02510
	M1=M+1	TUR02520
	MM1=MM+1	TUR02530
	MM2=MM+2	TUR02540
	MSG=SQRT(FLOAT(MM))	TUR02550
	DO 700 I=2,M1	TUR02560
	FR(I)=(I-1)*DELF	TUR02570
	700 CONTINUE	TUR02580
	R2=DIAM/THET	TUR02590
	R=RANG+R2	TUR02600
	DT=THET*RANG	TUR02610
	D2=DIAM+DT	TUR02620
C	COMPUTATION OF EFFECTIVE WIND VELOCITY, COHERENCE LENGTH AND	TUR02630
C	NORMALIZATION FREQUENCY FOR EACH SEGMENT OF PATH FROM LASER	TUR02640
C	DESIGNATOR TO TARGET	TUR02650
	ZI=0.0	TUR02660
	ZI=DELZ/2.	TUR02670
	ROT=0.0	TUR02680
	DEL=0.0	TUR02690
	DO 800 I=1,N1	TUR02700
	ZI=ZI+ZI	TUR02710
	ZI=DELZ	TUR02720
	VE1=V1(I)+TDOT*(ZI-R2)	TUR02730
	RO(I)=16.71*DELZ*CN1(I)*(1.-ZI/RANG)**1.66667/(LAMB*LAMB)	TUR02740
	FO(I)=VE1/(PI*D2*ZI/R)	TUR02750
	ROT=ROT+RO(I)	TUR02760
	DEL=DEL+DELZ*CN1(I)*(RANG-ZI)/RANG	TUR02770
	RO(I)=RO(I)**(-.6)	TUR02780
	800 CONTINUE	TUR02790
		TUR02800

```

      ROT=ROT**(-.6)
      CALL SPREAD(DIAM,ROT,LAMB,THET,RANG,DRO,THETDL,DDL,THET12,D12,
1      DTHET,D22)
      WRITE (IOOUT,4500) R2,R
      WRITE (IOOUT,4600) ROT,DRO,THET,DT,THETDL,DDL,THET12,D12,DTHET,D22
      WRITE (IOOUT,4700)
      DO 1000 I=1,N1
1000 WRITE (IOOUT,4800) I,RO(I),FO(I)
C  COMPUTATION OF ANGLE OF ARRIVAL POWER SPECTRUM OF LASER DESIGNATOR
      F2=0.
      PS(I)=0.
      DO 1100 J=2,M1
      F=FR(J)
      F1=0.
      CALL SPECT(F,D2,N1,F1,LAMB)
      PS(J)=F1
      IF (IOPT.EQ.1) PS(J)=F1*(D2/DIAM)**2
      F2=F2+PS(J)*DELF
1100 CONTINUE
      IF (IOPT.EQ.1) WRITE (IOOUT,4400) F2
      F2SQRT=SQRT(F2)
      AJITT=F2SQRT*RANG
      IF (IOPT.EQ.1) WRITE (IOOUT,4900) F2SQRT,AJITT
      IF (IOPT.EQ.1) GO TO 1500
C
      DTV=THET*R1V
C  COMPUTATION OF EFFECTIVE WIND VELOCITY, COHERENCE LENGTH AND
C  NORMALIZATION FREQUENCY FOR EACH SEGMENT OF PATH FROM
C  TARGET TO SEEKER
      ZI=0.
      ZI=DELIV/2.
      ROT=0.0
      DO 1200 I=1,N2
      ZI=ZI+Z1
      Z1=DELIV
      VEI=V2(I)+TDOT*(R1V-ZI)
      RO(I)=16.71*DELIV*CN2(I)*(ZI/R1V)**1.66667/(LAMB*LAMB)
      ROT=ROT+RO(I)
      RO(I)=RO(I)**(-.6)
      FO(I)=VEI/(PI*DIV*ZI/R1V)
1200 CONTINUE
      ROT=ROT**(-.6)
      CALL SPREAD(D22,ROT,LAMB,THET,R1V,DRO,THETDL,DDL,THET12,D12,DTHET,
1      D22V)
      WRITE (IOOUT,4550)
      WRITE (IOOUT,4600) ROT,DRO,THET,DTV,THETDL,DDL,THET12,D12,DTHET,
1      D22V
      WRITE (IOOUT,4700)
      DO 1300 I=1,N2
1300 WRITE (IOOUT,4800) I,RO(I),FO(I)
C  COMPUTATION OF TURBULENCE INDUCED POINTING JITTER POWER SPECTRUM
C  FROM TARGET SPOT TO LASER SEEKER, COMPUTATION OF TOTAL POWER
C  SPECTRUM FROM LASER DESIGNATOR TO SEEKER AND POWER SPECTRUM VARIANCE
      CALL THETO(THETA0,FALPH,CN2,D1V,R1V,N2)
      F2=0.
      DO 1400 J=2,M1
      F=FR(J)
      F1=0.
      CALL SPECT(F,D1V,N2,F1,LAMB)
      PS(J)=PS(J)+F1/(1.+(D2/(R1V*THETA0))**2)
      PS(J)=PS(J)*(D2/D1V)**2
      F2=F2+PS(J)*DELF
1400 CONTINUE
      WRITE (IOOUT,4400) F2
1500 IF (NPRINT.EQ.1.OR.NPRINT.EQ.3) WRITE (IOOUT,5000) DELF
      FREQ=0.0
      FINC=DELF*10.0
      IF (NPRINT.EQ.1.OR.NPRINT.EQ.3) WRITE (IOOUT,5600)
      IF (NPRINT.EQ.1.OR.NPRINT.EQ.3) WRITE (IOOUT,5800) PSCHAR
      K=1

```

```

TUR02810
TUR02820
TUR02830
TUR02840
TUR02850
TUR02860
TUR02870
TUR02880
TUR02890
TUR02900
TUR02910
TUR02920
TUR02930
TUR02940
TUR02950
TUR02960
TUR02970
TUR02980
TUR02990
TUR03000
TUR03010
TUR03020
TUR03030
TUR03040
TUR03050
TUR03060
TUR03070
TUR03080
TUR03090
TUR03100
TUR03110
TUR03120
TUR03130
TUR03140
TUR03150
TUR03160
TUR03170
TUR03180
TUR03190
TUR03200
TUR03210
TUR03220
TUR03230
TUR03240
TUR03250
TUR03260
TUR03270
TUR03280
TUR03290
TUR03300
TUR03310
TUR03320
TUR03330
TUR03340
TUR03350
TUR03360
TUR03370
TUR03380
TUR03390
TUR03400
TUR03410
TUR03420
TUR03430
TUR03440
TUR03450
TUR03460
TUR03470
TUR03480
TUR03490
TUR03500

```

```

1600 L=10
      IF (L.GT.M1) L=M1
      IF (NPRINT.EQ.1.OR.NPRINT.EQ.3) WRITE (IOOUT,2800) FREQ,(PS(J),J=K,L)
      FREQ=FREQ+FINC
      K=L+1
      L=K+9
      IF (K.LE.M1) GO TO 1600
      DO 2400 L=1,IOPT
      IF (L.EQ.1) WRITE (IOOUT,5100)
      IF (L.EQ.2) WRITE (IOOUT,5200)
C GENERATION OF RANDOM SEQUENCE HAVING SAME POWER SPECTRUM VARIANCE
C AS INDUCED BY TURBULANCE. ADD SYMMETRIC TERMS FOR NEGATIVE
C FREQUENCIES. COMPUTE MEAN AND VARIANCE OF RANDOM ARRAY
      RAN(1)=(0.,0.)
      DO 1700 I=2,M1
      MMM=MM2-I
C GENERATE RANDOM NUMBER WITH ROUTINE GAUSS
C NORMAL DISTRIBUTION
      MEAN = 0.0
      STANDARD DEVIATION = 1.0
      REAL PART = RANDOM NUMBER
      IMAG PART = 0
      RAN(I)=CMPLX(GAUSS(12,0.0,1.0),0.0)*SQRT(PS(I)/DELT)
      RAN(MMM)=RAN(I)
1700 CONTINUE
C CALCULATE POWER OF 2 ,NPOW, FOR FFT4 SINCE ARRAY PASSED TO
C FFT4 MUST HAVE SIZE THAT IS A POWER OF 2. (NOTE LN(2)=0.693147.)
      NPOW=IFIX(ALOG(FLOAT(MM)))/0.693147)
      NMAX=2**NPOW
      IF (MM.EQ.NMAX) GO TO 1900
C IF MM IS NOT A POWER OF 2 THEN RESET THE REST OF ARRAY RAN
      NPOW=NPOW+1
      ISTART=MM+1
      NMAX=2**NPOW
      DO 1800 I=ISTART,NMAX
      RAN(I)=(0.0,0.0)
1800 CONTINUE
C COMPUTE AND WRITE MEAN AND VARIANCE OF RANDOM ARRAY
1900 WRITE (IOOUT,5300)
      CALL MEANVR(1,M)
C FAST FOURIER TRANSFORM RANDOM ARRAY
C CDC ROUTINE CALL
      CALL FFT(RAN,MM,+1)
      CALL FFT4(1.0,RAN,NPOW,NMAX)
      DO 2100 I=1,MM
      RAN(I)=RAN(I)/MSQ
2100 CONTINUE
C COMPUTE AND WRITE MEAN AND VARIANCE OF TIME SEQUENCE.
      WRITE (IOOUT,5500)
      CALL MEANVR(M1,MM)
C WRITE TRANSFORMED ARRAY VALUES CORRESPONDING TO TIME VALUES
C OF POINTING JITTER FOR ONE DIRECTION.
      IF (NPRINT.EQ.2.OR.NPRINT.EQ.3) WRITE (IOOUT,5400) DELT
      DTIME=DELT
      TINC=DELT*10.0
      IF (NPRINT.EQ.2.OR.NPRINT.EQ.3) WRITE (IOOUT,5700)
      IF (NPRINT.EQ.2.OR.NPRINT.EQ.3) WRITE (IOOUT,5800) PJCHAR
      DO 2300 I1=M1,MM,10
      DO 2200 I2=1,10
      I3=I1+I2-1
      RR(I2)=REAL(RAN(I3))
2200 CONTINUE
      IF (NPRINT.EQ.2.OR.NPRINT.EQ.3) WRITE (IOOUT,2800) DTIME,
      1 (RR(I),I=1,10)
      DTIME=DTIME+TINC
2300 CONTINUE
2400 CONTINUE
C
2500 FORMAT (1X,100(1H*),/,13H WAVELENGTH (,F10.3,7H) GREAT,

```

```

TUR03510
TUR03511
TUR03512
TUR03540
TUR03550
TUR03560
TUR03570
TUR03580
TUR03590
TUR03600
TUR03610
TUR03620
TUR03630
TUR03640
TUR03650
TUR03660
TUR03670
TUR03680
TUR03690
TUR03700
TUR03710
TUR03720
TUR03730
TUR03740
TUR03750
TUR03760
TUR03770
TUR03780
TUR03790
TUR03800
TUR03810
TUR03820
TUR03830
TUR03840
TUR03850
TUR03860
TUR03870
TUR03880
TUR03890
TUR03900
TUR03910
TUR03920
TUR03930
TUR03940
TUR03950
TUR03960
TUR03970
TUR03980
TUR03990
TUR04000
TUR04010
TUR04020
TUR04030
TUR04040
TUR04050
TUR04060
TUR04070
TUR04080
TUR04090
TUR04100
TUR04110
TUR04120
TUR04130
TUR04140
TUR04150
TUR04160
TUR04170
TUR04180
TUR04190
TUR04200

```

```

1      40HER THAN 14 MICRONS: CONTROL RETURNED TO , TUR04210
2      21HMAIN FROM TURBULENCE.,/,1X,100(1H*)) TUR04220
2700 FORMAT (5F16.6) TUR04230
2800 FORMAT (E10.4,10E12.4) TUR04240
3000 FORMAT (A4,6X,7E10.4) TUR04250
3001 FORMAT(1H0,20X,45HTHE FOLLOWING ID-FIELD NOT RECOGNIZED BY TURB.,/ TUR04260
1      1X,A4,6X,7E10.4) TUR04270
3100 FORMAT (1H1,36H CALCULATION OF POWER SPECTRUM AND TUR04280
1      11HTURBULENCE 29HINDUCED POINTING JITTER OF A TUR04290
2      23HLASER TARGET DESIGNATOR) TUR04300
3200 FORMAT (1H+,101X,10HAND SEEKER) TUR04310
3300 FORMAT (1H0,41X,30HLASER WAVELENGTH (MICROMETERS),8X, TUR04320
1      F10.4,/,42X,26HDESIG. APERTURE DIAMETER ( TUR04330
2      7HMETERS),5X,F10.6,/,42X,18HBEAMSPREAD ANGLE ( TUR04340
3      8HRADIANS),12X,F10.6) TUR04350
3400 FORMAT (1H0,41X,33HSEEKER APERTURE DIAMETER (METERS),5X, TUR04360
1      F10.6,/,42X,36HRANGE FROM TARGET TO SEEKER (METERS), TUR04370
2      2X,F10.2) TUR04380
3500 FORMAT (1H0,41X,24HBEAM SLUE RATE (RAD/SEC),14X,F10.6, TUR04390
1      /,42X,26HDESIGNATION RANGE (METERS),12X, TUR04400
2      F10.2) TUR04410
3600 FORMAT (1H0,41X,26HDURATION OF TEST (SECONDS),12X, TUR04420
1      F10.4) TUR04430
3700 FORMAT (1H0,41X,30HTOTAL DESIGNATOR PATH SEGMENTS,15X, TUR04440
1      13) TUR04450
3800 FORMAT (1H0,41X,26HTOTAL SEEKER PATH SEGMENTS,19X,13) TUR04460
3900 FORMAT (1H0,41X,27HTOTAL FREQUENCIES FOR WHICH,/,42X, TUR04470
1      22H POWER SPECTRUM IS TO,/,42X, TUR04480
2      15H BE CALCULATED,27X,14) TUR04490
4000 FORMAT (1H0,/,26X,29H VALUES OF REFRACTIVE INDEX TUR04500
1      37HSTRUCTURE CONSTANT AND WIND SPEED IN TUR04510
2      15HDESIGNATOR PATH) TUR04520
4100 FORMAT (1H0,/,26X,29H VALUES OF REFRACTIVE INDEX TUR04530
1      37HSTRUCTURE CONSTANT AND WIND SPEED IN TUR04540
2      11HSEEKER PATH) TUR04550
4200 FORMAT (1H0,62X,5HCN**2,9X,9HWINDSPEED,16X,/,42X, TUR04560
1      11HSEGMENT NO.,5X,15H(METER**(-2/3)),3X, TUR04570
2      11H(METER/SEC),/) TUR04580
4300 FORMAT (1H ,46X,12,10X,E12.6,5X,F10.2) TUR04590
4400 FORMAT (1H0,/,36H THE VARIANCE OF THE POWER SPECTRUM TUR04600
1      3HIS ,E12.4) TUR04610
4500 FORMAT (1H1,/,56X,20HDESIGNATOR TO TARGET,/,38X, TUR04620
1      /,24X,41HVIRTUAL POINT SOURCE TO APERTURE DISTANCE, TUR04630
2      28X,F10.5,9H (METERS),/,24X,22HDISTANCE FROM VIRTUAL TUR04640
3      22HPOINT SOURCE TO TARGET,25X,F10.5,9H (METERS)) TUR04650
4550 FORMAT (1H1,/,58X,16HTARGET TO SEEKER,/) TUR04660
4600 FORMAT (1H0,23X,27HINTEGRATED COHERENCE LENGTH,42X,F10.6, TUR04670
1      9H (METERS),/,24X,36HDIAMETER/INTEGRATED COHERENCE LENGTHTUR04680
2      ,33X,F10.6,/,24X,31HTRANSMITTER-INDUCED BEAM SPREAD TUR04690
3      13X,E12.5,10H (RADIANS),3X,F10.6,9H (METERS), TUR04700
4      /,24X,31HDIFFRACTION-LIMITED BEAM SPREAD,13X,E12.5, TUR04710
5      10H (RADIANS),3X,F10.6,9H (METERS),/,24X, TUR04720
6      38HDIFFRACTION AND TURBULENCE BEAM SPREAD,6X,E12.5, TUR04730
7      10H (RADIANS),3X,F10.6,9H (METERS),/,24X,6HTOTAL TUR04740
8      19HEFFECTIVE BEAM SIZE,19X,E12.5,10H (RADIANS),3X,F10.6, TUR04750
9      9H (METERS),/) TUR04760
4700 FORMAT (1H0,34X,11HSEGMENT NO.,9X,16HCOHERENCE LENGTH, TUR04770
1      5X,27H REFERENCE FREQUENCY(HERTZ),/) TUR04780
4800 FORMAT (1H ,38X,12,10X,F16.6,10X,F16.6) TUR04790
4900 FORMAT (1H0,19H RMS SPOT JITTER = ,E10.4,10H RAD, OR = TUR04800
1      ,E10.4,7H METERS,/) TUR04810
5000 FORMAT (1H1,1X,40X,31H CALCULATED POWER SPECTRUM VS. TUR04820
1      9HFREQUENCY,/,47X,16H AT INTERVALS OF,F6.3,6H HERTZ) TUR04830
5100 FORMAT (1H0,37H OUTPUT FOR DESIGNATOR TO TARGET PATH) TUR04840
5200 FORMAT (1H0,33H OUTPUT FOR TARGET TO SEEKER PATH) TUR04850
5300 FORMAT (1H0,35X,35H MEAN AND VARIANCE OF RANDOM ARRAY TUR04860
1      /,/) TUR04870
5400 FORMAT (1H1,49X,34HVALUES OF POINTING JITTER VS. TIME / TUR04880
1      52X,16H AT INTERVALS OF,F8.4,4H SEC) TUR04890
5500 FORMAT (1H0,/,35X,28H MEAN AND VARIANCE OF TIME TUR04900

```

```

      8HSEQUENCE//)
5600 FORMAT (1H ,10H BEGINNING,/,10H FREQ (HZ))
5700 FORMAT (1H ,2X,5H TIME,/,2X,6H (SEC))
5800 FORMAT (1H+,T13,46(1H-),1H ,4A4,55(1H-),/)
      RETURN
      END

```

```

TUR04910
TUR04920
TUR04930
TUR04940
TUR04950
TUR04960

```

```

C  FUNCTION DESUB(X,DRO)
    FUNCTION USED FOR INTEGRATION F(X)
    FACTR=3.44*(DRO*X)**1.666667*(1.-X**0.333333)
    IF(FACTR.LT.160.) GO TO 10
    DESUB=0.
    GO TO 20
10  ARCCX=ATAN2(SQRT(1.-X**2),X)
    DESUB=X*((ARCCX-X*(1.-X**2)**.5)*EXP(-FACTR))
20  RETURN
    END

```

```

DES00010
DES00020
DES00030
DES00040
DES00050
DES00060
DES00070
DES00080
DES00090
DES00100

```



```

FUNCTION FALPH(XI1)
IF (XI1.GE..5623) GO TO 100
FALPH=10.66*((XI1)**2)
GO TO 900
100 IF (XI1.GE.1.0) GO TO 200
FALPH=4.025*XI1-.00659
GO TO 900
200 IF (XI1.GE.1.778) GO TO 300
FALPH=1.8547*XI1+2.164
GO TO 900
300 IF (XI1.GE.3.162) GO TO 400
FALPH=.8475*XI1+3.955
GO TO 900
400 IF (XI1.GE.5.623) GO TO 500
FALPH=.391*XI1+5.3977
GO TO 900
500 IF (XI1.GE.10.) GO TO 600
FALPH=.1814*XI1+6.578
GO TO 900
600 IF (XI1.GT.31.62) GO TO 700
FALPH=.0534*XI1+7.95
GO TO 900
700 IF (XI1.GT.1000.) GO TO 800
FALPH=7.8*(XI1**.06)
GO TO 900
800 FALPH=11.97
900 RETURN
END

```

```

FAL00010
FAL00020
FAL00030
FAL00040
FAL00050
FAL00060
FAL00070
FAL00080
FAL00090
FAL00100
FAL00110
FAL00120
FAL00130
FAL00140
FAL00150
FAL00160
FAL00170
FAL00180
FAL00190
FAL00200
FAL00210
FAL00220
FAL00230
FAL00240
FAL00250
FAL00260
FAL00270
FAL00280

```

	SUBROUTINE FFT4(SIGN,X,NPOW,NMAX)	FFT00010
	COOLEY-TUKEY METHOD OF FOURIER TRANSFORM	FFT00020
	INCLUDES SINE COSINE COMPUTATION AND	FFT00030
	REARRANGING DATA ACCORDING TO REVERSE BIT ADDRESSES	FFT00040
	SIGN = FOURIER DIRECTION TRANSFORM FLAG	FFT00050
	-1. FOR DIRECT TRANSFORM, TO COEFFICIENTS FROM SERIES	FFT00060
	1. FOR INVERSE TRANSFORM, TO SERIES FROM COEFFICIENTS	FFT00070
	X = LOC. OF FOURIER TRANSFORM BLOCK	FFT00080
	NPOW = POWER OF 2 TO OBTAIN NMAX	FFT00090
	NMAX = LENGTH OF BLOCK X	FFT00100
	COMPLEX X,CXCS,HOLD,XA	FFT00110
	DIMENSION CS(2),MSK(13)	FFT00120
	DIMENSION X(1)	FFT00130
	EQUIVALENCE (CXCS,CS)	FFT00140
	ZZ=6.283185306*SIGN/LOAT(NMAX)	FFT00150
	MSK(1)=NMAX/2	FFT00160
	DO 100 I=2,NPOW	FFT00170
	MSK(I)=MSK(I-1)/2	FFT00180
100	CONTINUE	FFT00190
	NN=NMAX	FFT00200
	MM=2	FFT00210
C	LOOP OVER NPOW LAYERS	FFT00220
	DO 800 LAYER=1,NPOW	FFT00230
	NN=NN/2	FFT00240
	NW=0	FFT00250
	DO 700 I=1,MM,2	FFT00260
	II=NN*I	FFT00270
C	CXCS = CEXP(2*PI*NW*SIGN/NMAX)	FFT00280
	W=FLOAT(NW)*ZZ	FFT00290
	CS(1)=COS(W)	FFT00300
	CS(2)=SIN(W)	FFT00310
C	COMPUTE ELEMENTS FOR BOTH HALFS OF EACH BLOCK	FFT00320
	DO 200 J=1,NN	FFT00330
	II=II+1	FFT00340
	IJ=II-NN	FFT00350
	XA=CXCS*X(II)	FFT00360
	X(II)=X(IJ)-XA	FFT00370
	X(IJ)=X(IJ)+XA	FFT00380
200	CONTINUE	FFT00390
C	BUMP UP SERIES BY 2	FFT00400
C	COMPUTE REVERSE ADDRESS	FFT00410
	DO 400 LOC=2,NPOW	FFT00420
	LL=NW-MSK(LOC)	FFT00430
	IF (LL) 500,600,300	FFT00440
300	NW=LL	FFT00450
400	CONTINUE	FFT00460
500	NW=MSK(LOC)+NW	FFT00470
	GO TO 700	FFT00480
600	NW=MSK(LOC+1)	FFT00490
700	CONTINUE	FFT00500
	MM=MM*2	FFT00510
800	CONTINUE	FFT00520
C	DO FINAL REARRANGEMENT	FFT00530
	NW=0	FFT00540
	DO 1600 I=1,NMAX	FFT00550
	NW1=NW+1	FFT00560
	HOLD=X(NW1)	FFT00570
	IF (NW1-1) 1100,1000,900	FFT00580
900	X(NW1)=X(I)	FFT00590
1000	X(I)=HOLD	FFT00600
C	BUMP UP SERIES BY 1	FFT00610
C	COMPUTE REVERSE ADDRESS	FFT00620
1100	DO 1300 LOC=1,NPOW	FFT00630
	LL=NW-MSK(LOC)	FFT00640
		FFT00650
		FFT00660
		FFT00670
		FFT00680
		FFT00690
		FFT00700

```

      IF (LL) 1400,1500,1200
1200  NW=LL
1300  CONTINUE
1400  NW=MSK(LOC)+NW
      GO TO 1600
1500  NW=MSK(LOC+1)
1600  CONTINUE
      IF (SIGN) 1900,1900,1700
1700  PTS=NMAX
      DO 1800 I=1,NMAX
      X(I)=X(I)/PTS
1800  CONTINUE
1900  RETURN
      END

```

```

FFT00710
FFT00720
FFT00730
FFT00740
FFT00750
FFT00760
FFT00770
FFT00780
FFT00790
FFT00800
FFT00810
FFT00820
FFT00830
FFT00840

```

```

C      FUNCTION GAUSS(N,XBAR,SIGMA)
C      GENERATE RANDOM NUMBERS WITH NORMAL DISTRIBUTION
C      MEAN = XBAR
C      STANDARD DEVIATION = SIGMA
      DATA NN /0/
      IF (NN.GT.0) GO TO 1
      NN=1
      C=0.
1     CONTINUE
      X=0.0
      IF (C.EQ.0.0) C=735.34829
      DO 100 J=1,N
      C=Rand(C)
      X=X+C
100    CONTINUE
      XN=N
      X=SQRT(12.0/XN)*(X-0.5*XN)
      GAUSS=SIGMA*X+XBAR
      RETURN
      END

```

```

GAU00010
GAU00030
GAU00040
GAU00050
GAU00060
GAU00070
GAU00080
GAU00090
GAU00100
GAU00110
GAU00120
GAU00130
GAU00140
GAU00150
GAU00160
GAU00170
GAU00180
GAU00190
GAU00200
GAU00210

```

```

SUBROUTINE MEANVR(N1,N2)
COMPUTE AND WRITE MEAN AND VARIANCE OF COMPLEX ARRAY
OVER SOME RANGE OF THE ARRAY.
  ++ INPUT ++
  N1 = STARTING INDEX OF RANGE
  N2 = ENDING INDEX OF RANGE
  ++ COMMON ++
  RAN = COMPLEX ARRAY CONTAINING DATA
  ++ OUTPUT ++
  MEAN1 = MEAN OF REAL PART
  MEAN2 = MEAN OF IMAGINARY PART
  VAR1 = VARIANCE OF REAL PART
  VAR2 = VARIANCE OF IMAGINARY PART
  COMPLEX RAN
  REAL MEAN1,MEAN2
  THIS IS A COMPLEX NUMBER EACH NUMBER TAKES TWO WORDS
  COMMON /M05/RAN(2048)
  COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLINT,KSTOR,NPLOTUM
  MEAN1=0.
  MEAN2=0.
  VAR1=0.
  VAR2=0.
  DO 100 I=N1,N2
  MEAN1=REAL(RAN(I))+MEAN1
  MEAN2=AIMAG(RAN(I))+MEAN2
  VAR1=(REAL(RAN(I)))**2+VAR1
  VAR2=(AIMAG(RAN(I)))**2+VAR2
100 CONTINUE
  MEAN1=MEAN1/FLOAT(N2-N1)
  MEAN2=MEAN2/FLOAT(N2-N1)
  VAR1=VAR1/FLOAT(N2-N1)
  VAR2=VAR2/FLOAT(N2-N1)
  WRITE (IOOUT,200) MEAN1,MEAN2
  WRITE (IOOUT,300) VAR1,VAR2
  RETURN
200 FORMAT (35X,20H MEAN OF REAL PART =,E12.5,10H, MEAN OF
1 11HIMAG PART =,E12.5)
300 FORMAT (35X,20H VAR. OF REAL PART =,E12.5,10H, VAR. OF
1 11HIMAG PART =,E12.5)
END

```

```

MEA00010
MEA00020
MEA00030
MEA00040
MEA00050
MEA00060
MEA00070
MEA00080
MEA00090
MEA00100
MEA00110
MEA00120
MEA00130
MEA00140
MEA00150
MEA00160
MEA00170
MEA00180
MEA00190
MEA00200
MEA00210
MEA00220
MEA00230
MEA00240
MEA00250
MEA00260
MEA00270
MEA00280
MEA00290
MEA00300
MEA00310
MEA00320
MEA00330
MEA00340
MEA00350
MEA00360
MEA00370
MEA00380
MEA00390
MEA00400
MEA00410
MEA00420
MEA00430
MEA00440
MEA00450
MEA00460
MEA00470
MEA00480
MEA00490
MEA00500
MEA00510
MEA00520
MEA00530

```

	SUBROUTINE SPECT(F,D2,N,F1,LAMB)	SPE00010
C	THIS ROUTINE GENERATES THE APPROXIMATE FUNCTION	SPE00020
C	G<ALPHA> (F/F<SUB 0,1>)	SPE00030
C	USED IN THE POWER SPECTRUM OF ANGLE-OF-ARRIVAL EQUATION	SPE00040
	REAL LAMB	SPE00050
	COMMON /MO1/FR(1025),CN(20),V1(20),FO(20),RO(20)	SPE00060
	FACT=1.32E-2*(LAMB/D2)**2	SPE00070
	F1=0.	SPE00080
	DO 100 I=1,N	SPE00090
	IF (F.LE..332*FO(I)) G=1.	SPE00100
	IF (F.GT..332*FO(I)) G=1.12-.361*F/FO(I)	SPE00110
	IF (F.GE.3.10*FO(I)) G=0.	SPE00120
	F1=F1+FACT*((D2/RO(I))**5/(F*F*FO(I))**.33333)*G	SPE00130
100	CONTINUE	SPE00140
	RETURN	SPE00150
	END	SPE00160

	SUBROUTINE SPREAD(DIAM,ROT,WAVE,THET,RANG,DRO,THETDL,DDL,THET12,	SPR00010
	1 D12,DTHET,DTOT)	SPR00020
C	COMPUTATION OF BEAM SPREAD ANGLE DUE TO DIFFRACTION AND DIFFRACTION	SPR00030
C	AND TURBULENCE AND SPOT DIAMETER ON TARGET (OR SEEKER).	SPR00040
C		SPR00050
	DRO=DIAM/ROT	SPR00060
C	DO 1/2 SIMPSON RULE INTEGRATION	SPR00070
	VARX=0.0	SPR00080
	DELTAX=0.01	SPR00090
	RDR0=DESUB(VARX,DRO)/2.0	SPR00100
	DO 100 I=1,100	SPR00110
	VARX=VARX+DELTAX	SPR00120
	RDR0=RDR0+DESUB(VARX,DRO)	SPR00130
100	CONTINUE	SPR00140
	RDR0=(RDR0-DESUB(VARX,DRO)/2.0)*DELTAX	SPR00150
	RDR0=1.0/(SQRT(5.092958*(DRO)**2*RDR0))	SPR00160
	THETDL=1.128*WAVE/DIAM	SPR00170
	DDL=THETDL*RANG	SPR00180
	THET12=THETDL*DRO*RDR0	SPR00190
	D12=THET12*RANG	SPR00200
	DTHET=SQRT(THET12**2+THET**2)	SPR00210
	DTOT=DIAM+DTHET*RANG	SPR00220
	RETURN	SPR00230
	END	SPR00240
		SPR00250

	SUBROUTINE THETO(THETA0,FALPH,CN2,D1V,R1V,N2)	THE00010
C	++ CALLED FUNCTIONS ++	THE00020
C	FALPH	THE00030
	DIMENSION CN2(20)	THE00040
	COMMON /IOUNIT,IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU	THE00050
	DEL1V=R1V/FLOAT(N2)	THE00060
C	CALCULATE D1INF	THE00070
	D1INF=0.	THE00080
	S1=DEL1V/2.	THE00090
	S=0.	THE00100
	D1V3=(D1V)*(-.3333)	THE00110
	DO 100 I=1,N2	THE00120
	S=S+S1	THE00130
	D1INF=DEL1V*CN2(I)*((S/R1V)**1.6667)+D1INF	THE00140
	S1=DEL1V	THE00150
100	CONTINUE	THE00160
	D1INF=0.5*11.97*D1V3*D1INF	THE00170
C	INITIAL ESTIMATE FOR THETA0	THE00180
C	WRITE (IOOUT,500) D1INF	THE00190
	THETA0=1.E-4	THE00200
200	XI1=0.	THE00210
	D1THE=0.	THE00220
	S=0.	THE00230
	S1=DEL1V/2.	THE00240
	DO 300 I=1,N2	THE00250
	S=S+S1	THE00260
	XI1=THETA0*(R1V-S)/D1V	THE00270
	D1THE=DEL1V*CN2(I)*((S/R1V)**1.6667)*FALPH(XI1)+D1THE	THE00280
	S1=DEL1V	THE00290
300	CONTINUE	THE00300
	D1THE=D1THE*D1V3	THE00310
	IF (ABS((D1INF-D1THE)/D1INF).LT..001) GO TO 400	THE00320
	THETA0=THETA0*(1+.5*(D1INF-D1THE)/D1INF)	THE00330
C	WRITE (IOOUT,500) THETA0,D1THE	THE00340
	GO TO 200	THE00350
400	CONTINUE	THE00360
C	WRITE (IOOUT,500) D1THE	THE00370
	RETURN	THE00380
C		THE00390
500	FORMAT (2E16.8)	THE00400
	END	THE00410

SUBROUTINE BASCAT(WAVE,EXCO,IERR)

THIS VERSION OF BASCAT (20 SEP 81) DIFFERS FROM THE EOSAEL 80 VERSION IN ITS INTERNAL PROGRAM STRUCTURE AND OUTPUT CAPABILITY. INPUT FORMATTING HAS NOT BEEN CHANGED. BRIEFLY, THE INTERNAL STRUCTURAL CHANGES CONSIST OF THE FOLLOWING :

- (A) SUBROUTINE THIT HAS BEEN ELIMINATED. THE FUNCTIONS WHICH IT ONCE PERFORMED HAVE BEEN CONSOLIDATED INTO SUBROUTINE START.
- (B) A NEW LIDAR BIASING ALGORITHM HAS BEEN INSERTED.
- (C) DIRECT BEAM (I.E., UNSCATTERED) COMPUTATIONS IN SUBROUTINE START HAVE BEEN REVISED.
- (D) SUBROUTINE CONV HAS BEEN MODIFIED SO THAT THE DIFFERENCE OF TWO NUMBERS RETAINS MORE SIGNIFICANT DIGITS.
- (E) NUMERICAL CHECKS FOR IMPROPER ARGUMENTS OF FUNCTIONS (DIVISIONS, SQUARE ROOTS, LOGARITHMS, ETC.) HAVE BEEN REVISED.
- (F) ARGUMENT LISTS OF A FEW COMMON BLOCKS HAVE BEEN CHANGED.
- (G) WRITE STATEMENTS FOR OUTPUTTING BASCAT RESULTS TO A USER-DEFINED PLOT FILE (NPLOTU) HAVE BEEN INCLUDED. THESE STATEMENTS MUST BE UNCOMMENTED IN ORDER TO ACTIVATE THEM.

THE BASCAT MODULE ALONE USES THE FOLLOWING SUBROUTINES:

- BKWD - CONTAINS BACKWARD SCATTERING ALGORITHM
- CONV - CONVOLVES IMPULSE RESPONSE WITH SQUARE PULSE
- ELM - DETERMINES BIASING DISTANCES
- FIND - DETERMINES INTERPOLATED PHASE FUNCTION VALUE
- FWRD - FIRST ORDER SCATTERING ALGORITHM
- GAS - DETERMINES MONTE CARLO SCATTERING ANGLES FOR TRAVERSES
- GMAX - DETERMINES MAXIMUM OF AN INPUT ARRAY
- MATRX - GENERATES ROTATION MATRICES
- ROTAT - ROTATES VECTORS FROM ONE COORDINATE SYSTEM TO ANOTHER
- SMOZZ - DETERMINES START OF TRAILING ZEROS IN INPUT ARRAY
- START - INITIATES PHOTON TRAJECTORIES
- TRAVRS - MOVES PHOTONS BETWEEN SCATTERING POINTS AND FINDS OBSERVED POWER CONTRIBUTIONS AT THOSE POINTS
- USCA - SELECTS RANDOM ANGLES WEIGHTED BY PHASE FUNCTION

TWO SUBROUTINES SHARED BY BASCAT WITH OTHER EOSAEL 80 MODULES ARE THE FOLLOWING :

- PFUNC - SELECTS AND RENORMALIZES PHASE FUNCTION DATA FROM EOSAEL 80 DATA BASE
- RAND - RANDOM NUMBER GENERATOR (GENERATES UNIFORM DISTRIBUTION OF RANDOM NUMBERS BETWEEN 0 AND 1)

** NOTE** THE FOLLOWING ROUTINES UTILIZE THE RANDOM NUMBER GENERATOR WHICH IS INVOKED AS FUNCTION 'RAND(SEED)' :

- (A) BKWD - 3 OCCURRENCES OF FUNCTION RAND
- (B) FWRD - 2 OCCURRENCES
- (C) GAS - 3 OCCURRENCES
- (D) START - 2 OCCURRENCES
- (E) TRAVRS - 2 OCCURRENCES

USERS OF OTHER (NON-MP) COMPUTER SYSTEMS MUST REPLACE FUNCTION RAND(SEED) WITH A UNIFORM RANDOM NUMBER GENERATOR WHICH WORKS FOR THEIR SYSTEMS. THE RANDOM NUMBER SEED IS INITIALIZED IN SUBROUTINE BASCAT. THIS AND ALL SUBSEQUENT VALUES OF THE RANDOM NUMBER SEED ARE PASSED VIA COMMON BLOCK 'RNDM'. SHOULD THE RANDOM NUMBER SEED USED HERE (735.34829) BE INAPPROPRIATE FOR THE USER'S SYSTEM, IT IS SUGGESTED THAT THE INITIALIZATION USED BELOW (SEED=735.34829) BE CHANGED TO AN APPROPRIATE VALUE.

SUBSTANTIAL MODIFICATIONS HAVE BEEN MADE TO THE FOLLOWING SUBROUTINES PRESENT IN EOSAEL 80 :

CONV

BAS0001
BAS0002
BAS0003
BAS0004
BAS0005
BAS0006
BAS0007
BAS0008
BAS0009
BAS0010
BAS0011
BAS0012
BAS0013
BAS0014
BAS0015
BAS0016
BAS0017
BAS0018
BAS0019
BAS0020
BAS0021
BAS0022
BAS0023
BAS0024
BAS0025
BAS0026
BAS0027
BAS0028
BAS0029
BAS0030
BAS0031
BAS0032
BAS0033
BAS0034
BAS0035
BAS0036
BAS0037
BAS0038
BAS0039
BAS0040
BAS0041
BAS0042
BAS0043
BAS0044
BAS0045
BAS0046
BAS0047
BAS0048
BAS0049
BAS0050
BAS0051
BAS0052
BAS0053
BAS0054
BAS0055
BAS0056
BAS0057
BAS0058
BAS0059
BAS0060
BAS0061
BAS0062
BAS0063
BAS0064
BAS0065
BAS0066
BAS0067
BAS0068
BAS0069
BAS0070

```

C      TRAVRS
C      START
C*****
COMMON/RNDM/SEED
COMMON/GEOMET/PTS(15),IGEOSW
COMMON/ALL/AT,BT,CT,BLIM
COMMON/IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLINT,KSTOR,NPLOTJ
COMMON/CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK
COMMON/MOS/V1,C1,S1,SA(3),EN(10,100),ENC(10),ELMIN,DELD,DTOT,
+NTMAX,NSCAM,KMAX,LMAX,LMM1
COMMON/BASPOT/UC(65),SUM(65),WVL(16),NWVL,ALBED(16),BS(16),
+BE(16),SINGWV,PF(65),LLMAX
COMMON/FAHT/SD(3),UV,GAMMA,ELD(2),STH,FAC,ALIM,THV,TAU,EL(2),
+ALB(2),ZG,DMAX
COMMON/FGEL/XA(3),D,NSCA
COMMON/FWD/AKM,R(3,3),AKSQ,XD(3),ASQ(3),RE(3,3),A(3)
COMMON/CONB/X(100),Y(100)
COMMON/HIT/UDS,THSP,RS(3,3),XS(3),DSA,XV(3)
DIMENSION NM(10),TPU(7),SS(3)
DIMENSION IAL(7),DAT(7),IOR(10)
DATA IAL/2HPA,2HSQ,2HDE,2HCL,2HGR,2HPU,2HGO /
DATA IZERO/0/
C*****
C      THIS SUBROUTINE CALCULATES STEADY STATE AND TIME-DEPENDENT DIRECT
C      AND MULTIPLY SCATTERED POWER INTO A DETECTOR BY AN ELLIPSOIDAL
C      AEROSOL CLOUD WITH GROUND PLANE, FOR A LASER SOURCE. THE DETECTOR
C      AND SOURCE MAY HAVE ANY LOCATIONS, LOOK ANGLES, AND CONE OF VIEW,
C      BEAM SPREAD/WAVELENGTH. THE AEROSOL CLOUD MAY HAVE ANY ORIENTA-
C      TION, SIZE, AND SCATTERING PHASE FUNCTION (ARBITRARY NORMALIZA-
C      TION), IN A COORDINATE SYSTEM WITH ORIGIN AT THE CLOUD CENTER,
C      WITH Z-AXIS VERTICAL, X-AXIS EAST, AND Y-AXIS NORTH. THE GROUND
C      PLANE, ASSUMED AN ISOTROPIC REFLECTOR, MAY HAVE ANY ALBEDO, AND
C      MAY OR MAY NOT INTERSECT THE AEROSOL CLOUD.
C*****
C** INPUT DATA CARDS ARE READ IN AN ORDER-INDEPENDENT MANNER, WITH
C** A FOUR-LETTER IDENTIFIER IN COLUMNS 1-4 OF EACH RECORD. DATA
C** ON EACH CARD IS READ IN UNDER THE FOLLOWING FORMAT :
C** (A4,1X,7(E9.4,1X)). NOTE THAT INTEGER VARIABLES IN THE PROGRAM
C** MUST BE INPUT AS REAL NUMBERS IN THIS INPUT SCHEME ... THEY ARE
C** LATER FIXED TO THE INTEGER TYPE.
C-----
C      CARD IDENTIFIER : PART
C      VARIABLES READ : N1,N2,ITIME
C      N1=NUMBER OF PARTIAL OUTPUTS DESIRED, FOR A GIVEN RUN
C      N2=NUMBER OF PHOTONS TO BE USED FOR EACH PARTIAL CALCULATION
C** NOTE ** FOR CERTAIN DETECTOR CONDITIONS, AND WITH NORMAL BIASING,
C      AS MANY AS 100,000 PHOTONS MAY BE NEEDED TO OVERCOME LARGE
C      STATISTICAL FLUCTUATIONS IN FIRST ORDER SCATTERING RETURNS.
C      SUCH CONDITIONS ARE DEFINED BY THE FOLLOWING CHARACTERISTICS
C      (A) THE DETECTOR IS IN A MONOSTATIC LIDAR CONFIGURATION.
C      (B) THE DETECTOR IS WITHIN 10 METERS OF THE CLOUD (OR IS INSIDE
C          OF IT).
C      (C) THE CLOUD IS NOT OPTICALLY THICK ALONG THE LOOK DIRECTION
C          (OPTICAL DEPTHS LESS THAN 10).
C*** IN ORDER TO ATTAIN MORE RAPID CONVERGENCE OF FIRST ORDER RETURN
C      POWER UNDER THE ABOVE CONDITIONS, A DIFFERENT BIASING SCHEME IS
C      USED. THE SPECIFIC CONDITIONS WHICH TRIGGER THIS ALTERNATE MODE
C      ARE THE FOLLOWING :
C      (A) THE DOT PRODUCT OF THE SOURCE APERTURE SURFACE NORMAL AND
C          THE DETECTOR APERTURE SURFACE NORMAL IS GREATER THAN 0.99.
C      (B) THE DISTANCE OF THE DETECTOR APERTURE FROM THE NEAREST
C          CLOUD SURFACE (AS SEEN ALONG THE DETECTOR NORMAL) IS LESS
C          THAN OR EQUAL TO 10 METERS (0.01 KM).
C      (C) THE SEPARATION OF SOURCE AND DETECTOR APERTURE CENTERS IS
C          LESS THAN OR EQUAL TO 10 TIMES THE DETECTOR APERTURE
C          RADIUS.

```

```

*** THE MAIN DIFFERENCES WHICH WILL BE OBSERVED BETWEEN THE NORMAL AND
    ALTERNATE BIASING MODES ARE THE FOLLOWING :
    (A) THE FIRST ORDER RETURN POWER IN THE EARLIEST TIME BOX WILL
        BE EXTREMELY STABLE IN THE ALTERNATE MODE.
    (B) ONLY THE EARLIEST FIRST ORDER SCATTERING RESULTS WILL BE
        STRONGLY AFFECTED.
    (C) LATER FIRST ORDER RETURNS AND ALL HIGHER ORDER RETURNS WILL
        HAVE NEGLIGIBLY POORER CONVERGENCE.
*** IN ORDER TO OBSERVE CONVERGENCE OF THE FIRST ORDER STEADY-STATE
    RETURN POWER TOWARD A STABLE VALUE, IT IS SUGGESTED THAT THE USER
    UTILIZE THE PARTIAL OUTPUT OPTION. AS AN EXAMPLE, IF 10,000
    PHOTONS ARE REQUIRED, SET N1=10 AND N2=1,000. THIS SELECTION WILL
    RUN 10 X 1,000 = 10,000 PHOTONS AND WILL OUTPUT RETURN POWER
    RESULTS AFTER EACH BATCH OF 1,000 PHOTONS.

    ITIME=OVERALL RUN NUMBER FOR THIS SET OF PARAMETERS. FOR EXAMPLE,
    ITIME=1 MEANS THE FIRST RUN, ITIME=3 MEANS THAT THE RESULTS
    OF THE TWO PREVIOUS RUNS WILL BE COMBINED WITH THIS 3RD RUN.

-----
CARD IDENTIFIER : SORC
VARIABLES READ : XS(1),XS(2),XS(3),THES,PHIS,ASMM
(XS(K),K=1,3)=SOURCE XYZ COORDINATES(KM)
THES,PHIS=(POLAR,AZIMUTHAL)ANGLES(DEG) OF SOURCE BEAM AXIS
ASMM=RADIUS OF SOURCE APERTURE(MM). THE SOURCE BEAM SPREAD ANGLE
    THSP IS TAKEN BY THE PROGRAM AS THE DIFFRACTION LIMIT FOR
    THIS APERTURE. IF YOU SET ASMM=0, THE PROGRAM PUTS THSP=0.
IF YOUR SOURCE PHOTONS WOULD NOT INTERSECT THE CLOUD OR THE
GROUND, THE SUBROUTINE NOTIFIES YOU, AND RETURNS.

-----
CARD IDENTIFIER : DETR
VARIABLES READ : XD(1),XD(2),XD(3),THED,PHID,THEV,ACM
(XD(K),K=1,3)=DETECTOR XYZ COORDINATES(KM)
THED,PHID=(POLAR,AZIMUTHAL)ANGLES(DEG) OF DETECTOR LOOK AXIS
THEV=DETECTOR CONE OF VIEW HALF-ANGLE(DEG)
ACM=RADIUS OF DETECTOR DISK(CM)
IF YOUR DETECTOR POINTS SKYWARD, OR IF NEITHER YOUR SOURCE NOR
YOUR DETECTOR LOOK INTO THE CLOUD, THE SUBROUTINE NOTIFIES YOU,
AND RETURNS.

-----
CARD IDENTIFIER : CLDS
VARIABLES READ : A(1),A(2),A(3),THE,PHE,PSE,ISO
(A(K),K=1,3)=ELLIPSOIDAL CLOUD PRINCIPAL HALF-AXES(KM)
THE,PHE,PSE=ELLIPSOID EULER ANGLES(DEG), WHERE 'PHE' IS THE FIRST
    ROTATION, ABOUT THE Z-AXIS, 'THE' IS THE NEXT, ABOUT THE NEW
    Y-AXIS, 'PSE' IS THE LAST, ABOUT THE NEW Z-AXIS.
ISO=AEROSOL IDENTIFICATION NUMBER, TO COMPARE WITH THE 'ID'
    PARAMETER.

-----
CARD IDENTIFIER : GRND
VARIABLES READ : ZG,ALBG
ZG=Z-COORDINATE OF GROUND PLANE(KM)
ALBG=GROUND PLANE REFLECTIVITY 0.<ALBG<1.
IF YOUR ZG IS SUCH THAT THE GROUND PLANE IS ENTIRELY ABOVE THE
CLOUD, THE SUBROUTINE INFORMS YOU, AND RETURNS. IF YOUR ZG IS
NEGATIVE, AND SO LARGE THAT NO GROUND REFLECTIONS WILL RETURN TO
THE DETECTOR WITHIN THE TIME LIMIT SET BY THE SUBROUTINE, THE SUB-
ROUTINE SETS ALBG=0. IF YOUR SOURCE IS BELOW THE GROUND PLANE,
THE SUBROUTINE PUTS THE SOURCE ON THE GROUND PLANE, AT YOUR XY
COORDINATES, AND NOTIFIES YOU.

-----
CARD IDENTIFIER : PULS
VARIABLES READ : TPU(1),TPU(2),...,TPU(7)
(TPU(J),J=1,7)=SOURCE PULSE DURATIONS(USEC)
YOU CAN INPUT AS MANY AS SIX DIFFERENT PULSE LENGTHS. THE LAST
ENTRY MUST BE BLANK(ZERO). THE SUBROUTINE CONVOLUTES THE MONTE
CARLO PROBABILITY PER UNIT TIME DATA WITH EACH OF THESE SQUARE
PULSES, AND, FOR EACH PULSE, WRITES THE TIME-DEPENDENT POWER TO
THE DETECTOR, FOR UNIT SOURCE PULSE POWER, FOR EACH SIGNIFICANT
ORDER OF MULTIPLE SCATTERING, AND FOR THE TOTAL OF ALL ORDERS.

```

C	CARD IDENTIFIER : GO	BAS02110
CC	VARIABLES READ : NONE	BAS02120
C	RUN TERMINATION CARD (MUST BE LAST CARD READ).	BAS02130
C	-----	BAS02140
C	IERR=0	BAS02150
C***	THE FOLLOWING STATEMENT MAY NEED TO BE CHANGED OR ELIMINATED	BAS02160
C***	FOR USE ON NON-HP1000 COMPUTERS.	BAS02170
C	SEED0=735.34829	BAS02180
	NDIM=100	BAS02190
	ND1=NDIM-1	BAS02200
	DTAUM=0.2	BAS02210
	GAMMA=EXCO	BAS02220
	SINGWV=WAVE	BAS02230
C	*** INPUT DATA INITIALIZATIONS.	BAS02240
C	IF(IZERO.NE.0) GO TO 600	BAS02250
	N1=0	BAS02260
	N2=0	BAS02270
	ITIME=0	BAS02280
	IS0=0	BAS02290
	DO 599 LL=1,3	BAS02300
	XS(LL)=0.	BAS02310
	XD(LL)=0.	BAS02320
	A(LL)=0.	BAS02330
	TPU(LL)=0.	BAS02340
599	TPU(LL+3)=0.	BAS02350
	TPU(7)=0.	BAS02360
	THES=0.	BAS02370
	PHIS=0.	BAS02380
	ASMM=0.	BAS02390
	THED=0.	BAS02400
	PHID=0.	BAS02410
	THEV=0.	BAS02420
	ACH=0.	BAS02430
	THE=0.	BAS02440
	PHE=0.	BAS02450
	PSE=0.	BAS02460
	ZG=0.	BAS02470
	ALBC=0.	BAS02480
	LLMAX=65	BAS02490
	IZERO=1	BAS02500
600	CONTINUE	BAS02510
C	*** READ BASCAT DATA SET RECORDS UNDER CARD-INDEPENDENT FORMAT	BAS02520
C	DO 700 K=1,7	BAS02530
	READ(IOIN,610) IA,IA1,(DAT(I),I=1,7)	BAS02540
610	FORMAT(2A2,1X,7(E9.4,1X))	BAS02550
	DO 615 JJ=1,8	BAS02560
	IF(IA.NE.IAL(JJ)) GO TO 615	BAS02570
	INOPT=JJ	BAS02580
	IF(INOPT.EQ.7) GO TO 701	BAS02590
	GO TO 620	BAS02600
615	CONTINUE	BAS02610
	IF((K.EQ.7).AND.(JJ.EQ.8)) GO TO 697	BAS02620
	GO TO 695	BAS02630
620	CONTINUE	BAS02640
	GO TO (621,622,623,624,625,626),INOPT	BAS02650
621	N1=IFIX(DAT(1))	BAS02660
	N2=IFIX(DAT(2))	BAS02670
	ITIME=IFIX(DAT(3))	BAS02680
	GO TO 700	BAS02690
622	XS(1)=DAT(1)	BAS02700
	XS(2)=DAT(2)	BAS02710
	XS(3)=DAT(3)	BAS02720
	THES=DAT(4)	BAS02730
	PHIS=DAT(5)	BAS02740
		BAS02750
		BAS02760
		BAS02770
		BAS02780
		BAS02790
		BAS02800

ASMM=DAT(6)	BAS02810
GO TO 700	BAS02820
623 XD(1)=DAT(1)	BAS02830
XD(2)=DAT(2)	BAS02840
XD(3)=DAT(3)	BAS02850
THED=DAT(4)	BAS02860
PHID=DAT(5)	BAS02870
THEV=DAT(6)	BAS02880
ACM=DAT(7)	BAS02890
GO TO 700	BAS02900
624 A(1)=DAT(1)	BAS02910
A(2)=DAT(2)	BAS02920
A(3)=DAT(3)	BAS02930
THE=DAT(4)	BAS02940
PHE=DAT(5)	BAS02950
PSE=DAT(6)	BAS02960
ISO=IFIX(DAT(7))	BAS02970
GO TO 700	BAS02980
625 ZG=DAT(1)	BAS02990
ALBG=DAT(2)	BAS03000
GO TO 700	BAS03010
626 CONTINUE	BAS03020
DO 627 NN=1,7	BAS03030
627 TPU(NN)=DAT(NN)	BAS03040
GO TO 700	BAS03050
C	BAS03060
C*** ERROR RETURNS	BAS03070
C	BAS03080
695 CONTINUE	BAS03090
WRITE(IOOUT,696)	BAS03100
696 FORMAT(1H0,20X,89H***BASCAT ERROR*** INPUT CARD DETECTED WHICH DOES	BAS03110
+S NOT MATCH ANY CORPECT INPUT IDENTIFIERS /)	BAS03120
IERR=1	BAS03130
GO TO 777	BAS03140
697 CONTINUE	BAS03150
WRITE(IOOUT,698)	BAS03160
698 FORMAT(1H0,20X,66H***BASCAT ERROR*** TOO MANY INPUT CARDS OR GO SE	BAS03170
+NTINEL NOT PRESENT /)	BAS03180
700 CONTINUE	BAS03190
701 CONTINUE	BAS03200
C	BAS03210
C*** GEOMETRICAL OPTION DATA TRANSFER	BAS03220
C	BAS03230
IF(IGEOSW.NE.1) GO TO 111	BAS03240
DO 110 I=1,3	BAS03250
XS(I)=PTS(I+6)-PTS(I+12)	BAS03260
110 XD(I)=PTS(I+3)-PTS(I+12)	BAS03270
111 CONTINUE	BAS03280
C	BAS03290
C*** GENERATE INTERPOLATED, RENORMALIZED PHASE FUNCTION	BAS03300
C	BAS03310
CALL PFUNC(ISO)	BAS03320
IF(GAMMA.EQ.0.0) GAMMA=BE(1)	BAS03330
ALBEDO=ALBED(1)	BAS03340
REWIND IPHFUN	BAS03350
ALB(1)=ALBEDO	BAS03360
ALB(2)=ALBG	BAS03370
C	BAS03380
C*** DETERMINE POWER OF 2 (KMAX) CORRESPONDING TO NUMBER OF PHASE	BAS03390
C*** FUNCTION VALUES PRESENT.	BAS03400
C	BAS03410
LMAX=LLMAX	BAS03420
LMM1=LMAX-1	BAS03430
KMAX=IFIX(ALOG(FLOAT(LMM1)))/0.693147)	BAS03440
C	BAS03450
C*** CC = SPEED OF LIGHT (KM/MICROSECOND)	BAS03460
C	BAS03470
CC=0.3	BAS03480
THV=PIRAD*THEV	BAS03490
IF(THV.LE.1.E-30) THV=0.	BAS03500

```

      UV=COS(THV)
C
C*** SET RANDOM NUMBER SEED
C
      SEED=(2*ITIME-1)*SEEDO
      AKM=ACM*1.E-5
      AKSQ=AKM**2
C
C*** SET MINIMUM DISTANCE ALIM (AND ITS SQUARE BLIM) SEPARATING
C*** THE DETECTOR DISK AND A PHOTON SCATTERING POINT.
C
      ALIM=AKM*10.
      BLIM=ALIM**2
      FAC=THV*AKSQ/8.
      BT=.61*WAVE*1.E-3
      THSP=0.
C
C*** DETERMINE DIFFRACTION-LIMITED SOURCE BEAMSPREAD
C
      IF(ASMM.GT.0.)THSP=BT/ASMM
C
C*** GENERATE ROTATION MATRIX RE( ) FOR CONVERSION FROM STANDARD
C*** FRAME OF REFERENCE TO CLOUD FRAME OF REFERENCE
C
      CALL MATRX(THV,PHE,SS,R)
      AT=PIRAD*PSE
      IF(AT.LE.1.E-30) AT=0.
      DO 1 I=1,3
      DO 1 J=1,3
1      RS(I,J)=0.
      RS(1,1)=COS(AT)
      RS(1,2)=SIN(AT)
      RS(2,1)=-RS(1,2)
      RS(2,2)=RS(1,1)
      RS(3,3)=1.
      DO 2 I=1,3
      DO 2 J=1,3
      RE(I,J)=0.
      DO 2 K=1,3
2      RE(I,J)=RE(I,J)+RS(I,K)*R(K,J)
C
C*** GENERATE ROTATION MATRIX RS( ) FOR CONVERSION FROM SOURCE
C*** CONE FRAME OF REFERENCE TO STANDARD FRAME OF REFERENCE
C
      CALL MATRX(THS,PHIS,SS,RS)
C
C*** ECHO INPUT PARAMETERS
C
      WRITE (IOOUT,6800)
      WRITE (IOOUT,4800)
      WRITE (IOOUT,4900)
      WRITE (IOOUT,5000)
      WRITE (IOOUT,4800)
      WRITE (IOOUT,4700)
      WRITE (IOOUT,5100)
      IF (ISO.EQ.0) WRITE (IOOUT,5200)
      IF (ISO.EQ.1) WRITE (IOOUT,5201)
      IF (ISO.EQ.2) WRITE (IOOUT,5202)
      IF (ISO.EQ.3) WRITE (IOOUT,5203)
      IF (ISO.EQ.4) WRITE (IOOUT,5204)
      IF (ISO.EQ.5) WRITE (IOOUT,5205)
      IF (ISO.EQ.6) WRITE (IOOUT,5206)
      IF (ISO.EQ.7) WRITE (IOOUT,5207)
      IF (ISO.EQ.8) WRITE (IOOUT,5208)
      IF (ISO.EQ.9) WRITE (IOOUT,5209)
      IF (ISO.EQ.10) WRITE (IOOUT,5210)
      IF (ISO.EQ.11) WRITE (IOOUT,5211)
      IF (ISO.EQ.12) WRITE (IOOUT,5212)
      WRITE (IOOUT,5600) WAVE,ALBEDO
      WRITE (IOOUT,5700) GAMMA

```

```

BAS03510
BAS03520
BAS03530
BAS03540
BAS03550
BAS03560
BAS03570
BAS03580
BAS03590
BAS03600
BAS03610
BAS03620
BAS03630
BAS03640
BAS03650
BAS03660
BAS03670
BAS03680
BAS03690
BAS03700
BAS03710
BAS03720
BAS03730
BAS03740
BAS03750
BAS03760
BAS03770
BAS03780
BAS03790
BAS03800
BAS03810
BAS03820
BAS03830
BAS03840
BAS03850
BAS03860
BAS03870
BAS03880
BAS03890
BAS03900
BAS03910
BAS03920
BAS03930
BAS03940
BAS03950
BAS03960
BAS03970
BAS03980
BAS03990
BAS04000
BAS04010
BAS04020
BAS04030
BAS04040
BAS04050
BAS04060
BAS04070
BAS04080
BAS04090
BAS04100
BAS04110
BAS04120
BAS04130
BAS04140
BAS04150
BAS04160
BAS04170
BAS04180
BAS04190
BAS04200

```

```

WRITE(IQOUT,5601)
WRITE(IQOUT,5602)
WRITE(IQOUT,5603)
WRITE(IQOUT,5701)
WRITE(IQOUT,5702)(XS(K),K=1,3)
WRITE(IQOUT,5704)THES
WRITE(IQOUT,5705)PHIS
WRITE(IQOUT,5703)ASMM
WRITE(IQOUT,5706)THSP
WRITE(IQOUT,5900)
WRITE(IQOUT,6000)THEV
WRITE(IQOUT,6100)ACM
WRITE(IQOUT,6201)(XD(K),K=1,3)
WRITE(IQOUT,6400)THED
WRITE(IQOUT,6500)PHID
WRITE(IQOUT,6501)
WRITE(IQOUT,6502)ZG,ALBG
DO 10 K=1,3
10 ASQ(K)=A(K)**2
DO 12 K=1,3
12 Y(K)=A(K)*2.*GAMMA
C
C*** DETERMINE LARGEST OPTICAL DEPTH PRESENT IN AEROSOL CLOUD
C
CALL GMAX(3,TAU)
C
C*** SET TIME AND DISTANCE INCREMENTS AND LIMITS
C
NTMAX=50
D=TAU/GAMMA
DELD=5.*D/NTMAX
DELT=DELD/CC
DMAX=5.5*D
C
C*** BEGIN CLOUD SUBBLOCK
C
C*****
C
THE FOLLOWING BLOCK OF WRITE STATEMENTS MUST BE UNCOMMENTED
IN ORDER TO OUTPUT DATA TO A USER-DEFINED PLOT FILE (NPLOTU).
C
THE OUTPUT QUANTITIES IN THIS BLOCK ARE THE FOLLOWING :
C
WAVE = WAVELENGTH (MICROMETERS)
C
ISO = AEROSOL TYPE (VALID RANGE, 0-12)
C
TAU = OPTICAL DEPTH ALONG LONGEST AXIS OF CLOUD ELLIPSOID
C
N1 = NUMBER OF PARTIAL RUNS WITHIN THIS BASCAT RUN
C
XS( ) = SOURCE XYZ POSITION ARRAY (KILOMETERS)
C
THES = SOURCE VECTOR POLAR ANGLE (DEGREES)
C
PHIS = SOURCE VECTOR AZIMUTHAL ANGLE (DEGREES)
C
ASMM = SOURCE APERTURE RADIUS (MILLIMETERS)
C
THSP = HALF-ANGLE OF SOURCE DIFFRACTION CONE (RADIAN)
C
XD( ) = DETECTOR XYZ POSITION ARRAY (KILOMETERS)
C
THED = DETECTOR VECTOR POLAR ANGLE (DEGREES)
C
PHID = DETECTOR VECTOR AZIMUTHAL ANGLE (DEGREES)
C
ACM = DETECTOR APERTURE RADIUS (CENTIMETERS)
C
THEV = HALF-ANGLE OF DETECTOR FIELD OF VIEW (DEGREES)
C
AK( ) = CLOUD ELLIPSOID PRINCIPAL HALF-AXIS ARRAY (KILOMETERS)
C
ALB(1)= SINGLE-SCATTERING ALBEDO OF CLOUD AEROSOL
C
ALB(2)= ALBEDO OF GROUND PLANE
C
WRITE(NPLOTU,9111) WAVE,ISO,TAU,N1
C9111 FORMAT(E9.4,1X,12,1X,E9.4,1X,2(12,1X))
C
WRITE(NPLOTU,9222)XS(1),XS(2),XS(3),THES,PHIS,ASMM,THSP
C
WRITE(NPLOTU,9222)XD(1),XD(2),XD(3),THED,PHID,ACM,THEV
C9222 FORMAT(12(E9.4,1X))
C
WRITE(NPLOTU,9222) A(1),A(2),A(3),ALB(1),ALB(2)
C*****
C
WRITE(IQOUT,6900)
WRITE(IQOUT,7000)(A(K),K=1,3)
WRITE(IQOUT,7100)THE,PHE,PSE

```

```

BAS04210
BAS04220
BAS04230
BAS04240
BAS04250
BAS04260
BAS04270
BAS04280
BAS04290
BAS04300
BAS04310
BAS04320
BAS04330
BAS04340
BAS04350
BAS04360
BAS04370
BAS04380
BAS04390
BAS04400
BAS04410
BAS04420
BAS04430
BAS04440
BAS04450
BAS04460
BAS04470
BAS04480
BAS04490
BAS04500
BAS04510
BAS04520
BAS04530
BAS04540
BAS04550
BAS04560
BAS04570
BAS04580
BAS04590
BAS04600
BAS04610
BAS04620
BAS04630
BAS04640
BAS04650
BAS04660
BAS04670
BAS04680
BAS04690
BAS04700
BAS04710
BAS04720
BAS04730
BAS04740
BAS04750
BAS04760
BAS04770
BAS04780
BAS04790
BAS04800
BAS04810
BAS04820
BAS04830
BAS04840
BAS04850
BAS04860
BAS04870
BAS04880
BAS04890
BAS04900

```

C	WRITE (IOOUT,7200)(Y(K),K=1,3)	BAS04910
C***	GENERATE ROTATION MATRIX R() FOR CONVERSION FROM DETECTOR	BAS04920
C***	CONE FRAME OF REFERENCE TO STANDARD FRAME OF REFERENCE	BAS04930
C	CALL MATRX(THED,PHID,SD,R)	BAS04940
C		BAS04950
C***	SET LIMIT NSCAM ON HIGHEST SIGNIFICANT SCATTERING ORDER (NSCAM-1)	BAS04960
C		BAS04970
	NSCAM=3.0*ALBEDO*TAU+2.0	BAS04980
	IF (NSCAM.GT.10) NSCAM=10	BAS04990
	IF (NSCAM.LT.3) NSCAM=3	BAS05000
	NSCA1=NSCAM-1	BAS05010
	AT=0.	BAS05020
	DO 4 K=1,3	BAS05030
4	AT=AT+RE(K,3)**2/ASQ(K)	BAS05040
	BT=1.-AT*ZG	BAS05050
	IF (BT.LE.0.) GO TO 4002	BAS05060
	AT=ZG+D	BAS05070
	IF (AT.GT.0.) GO TO 3	BAS05080
	ZG=-D	BAS05090
	WRITE(IOOUT,9004)ZG	BAS05100
3	CONTINUE	BAS05110
	IF (XS(3).GE.ZG) GO TO 16	BAS05120
	XS(3)=ZG	BAS05130
	WRITE(IOOUT,9003)XS(3)	BAS05140
16	CONTINUE	BAS05150
C		BAS05160
C***	DETERMINE DISTANCES ELD(1), ELD(2) FROM DETECTOR TO NEAREST AND	BAS05170
C***	FARTHEST CLOUD BOUNDARIES ALONG DETECTOR AXIS	BAS05180
C		BAS05190
	CALL ELM(XD,SD,ELD)	BAS05200
C		BAS05210
C***	DETERMINE DISTANCES EL(1), EL(2) FROM SOURCE TO NEAREST AND	BAS05220
C***	FARTHEST CLOUD BOUNDARIES ALONG SOURCE AXIS	BAS05230
C		BAS05240
	CALL ELM(XS,SS,EL)	BAS05250
C		BAS05260
C***	PERFORM GEOMETRICAL ERROR CHECKS	BAS05270
C		BAS05280
	IF ((EL(2).LE.0.) .AND. (ELD(2).LE.0.)) GO TO 4003	BAS05290
	IF ((EL(2).GT.0.) .AND. (ELD(2).GT.0.)) GO TO 6	BAS05300
	IF (EL(2).LE.0.) GO TO 7	BAS05310
	IF (ALB(2).LE.0.) GO TO 4004	BAS05320
	IF (SD(3).GE.0.) GO TO 4001	BAS05330
	AT=EL(1)	BAS05340
	CT=(ZG-XD(3))/SD(3)	BAS05350
	GO TO 9	BAS05360
7	CONTINUE	BAS05370
	IF (ALB(2).LE.0.) GO TO 4005	BAS05380
	IF (SS(3).GT.0.) GO TO 4000	BAS05390
	AT=(ZG-XS(3))/SS(3)	BAS05400
	GO TO 8	BAS05410
6	AT=EL(1)	BAS05420
C		BAS05430
C***	DETERMINE MINIMUM POSSIBLE TRAVERSE DISTANCE ELMIN	BAS05440
C		BAS05450
	8 CT=ELD(1)	BAS05460
	9 BT=0.	BAS05470
	DO 11 K=1,3	BAS05480
	X(K)=XS(K)+AT*SS(K)	BAS05490
	Y(K)=XD(K)+CT*SD(K)	BAS05500
11	BT=BT+(X(K)-Y(K))**2	BAS05510
	IF (BT.LE.1.E-30) BT=0.	BAS05520
	ELMIN=SQRT(BT)+AT+CT	BAS05530
	IF (ELMIN.LT.ALIM) ELMIN=ALIM	BAS05540
C		BAS05550
C***	DETERMINE VIRTUAL SOURCE POINT XV()	BAS05560
C		BAS05570
	DSA=0.	BAS05580
		BAS05590
		BAS05600


```

IF(ASMM.GT.0.)DSA=ASMM*1.E-6/TAN(THSP)
DO 305 K=1,3
305 XV(K)=XS(K)-DSA*SS(K)
C
C*** DETERMINE WHETHER NEAR-CLOUD LIDAR BIASING IS NEEDED.
C
DOTSD=0.
DOTXD=0.
DO 47 KL=1,3
DOTXD=DOTXD+(XD(KL)-XS(KL))*2
47 DOTSD=DOTSD+SS(KL)*SD(KL)
IF(DOTXD.GT.BLIM) DOTSD=0.
REWIND KSTOR
C
C*** START PHOTON LOOPS
C
DO 1000 II=1,N1
C
C*** CHECK TO SEE IF THIS IS THE FIRST RUN WITH NO PRIOR RESULTS USED
C
IF((II.EQ.1).AND.(ITIME.EQ.1))GO TO 1303
C
C*** RELOAD WITH RESULTS FROM PREVIOUS PARTIAL RUN
C
DO 1301 NSCA=1,NSCAM
READ(KSTOR)ENC(NSCA)
DO 1301 II=1,NTMAX
READ(KSTOR)EN(NSCA,II)
1301 CONTINUE
REWIND KSTOR
GO TO 1302
1303 CONTINUE
C
C*** INITIALIZATION FOR FIRST RUN
C
DO 1300 NSCA=1,NSCAM
ENC(NSCA)=0.
DO 1300 NT=1,NTMAX
1300 EN(NSCA,NT)=0.0
1302 CONTINUE
C
C*** LOAD EXPECTED CUMULATIVE NUMBER OF PHOTONS
C
DO 1304 NS=1,NSCAM
1304 ENC(NS)=ENC(NS)+N2
C
C*** DETERMINE VECTOR FROM VIRTUAL SOURCE POINT TO DETECTOR
C
DO 555 K=1,3
555 Y(K)=XD(K)-XV(K)
C
C*** START INNER PHOTON LOOP
C
DO 900 I2=1,N2
C
C*** INITIALIZE PHOTON DIRECTION AND CALCULATE DIRECT BEAM
C*** CONTRIBUTIONS (IF ANY)
C
CALL START(SS)
IF(STH.GT.0.)GO TO 150
GO TO 900
150 NSCA=1
C
C*** START MULTIPLE SCATTERING LOOP
C
I2FLG=0
ICOND=0
1700 NSCA=NSCA+1
C
C*** NEAR-CLOUD LIDAR BIASING IS ACTIVE WHEN ICOND=1. 10 PERCENT OF ALL

```

```

BAS05610
BAS05620
BAS05630
BAS05640
BAS05650
BAS05660
BAS05670
BAS05680
BAS05690
BAS05700
BAS05710
BAS05720
BAS05730
BAS05740
BAS05750
BAS05760
BAS05770
BAS05780
BAS05790
BAS05800
BAS05810
BAS05820
BAS05830
BAS05840
BAS05850
BAS05860
BAS05870
BAS05880
BAS05890
BAS05900
BAS05910
BAS05920
BAS05930
BAS05940
BAS05950
BAS05960
BAS05970
BAS05980
BAS05990
BAS06000
BAS06010
BAS06020
BAS06030
BAS06040
BAS06050
BAS06060
BAS06070
BAS06080
BAS06090
BAS06100
BAS06110
BAS06120
BAS06130
BAS06140
BAS06150
BAS06160
BAS06170
BAS06180
BAS06190
BAS06200
BAS06210
BAS06220
BAS06230
BAS06240
BAS06250
BAS06260
BAS06270
BAS06280
BAS06290
BAS06300

```

```

C**** FIRST ORDER PHOTONS USE AN ALTERNATE MODE OF BIASING IN THIS      BAS06310
C**** SITUATION. THE REMAINING 90 PERCENT ARE NORMALLY BIASED IN ALL    BAS06320
C**** ORDERS.                                                            BAS06330
C                                                                           BAS06340
C      IF((NSCA.NE.2).OR.(DOTSD.LE.0.99).OR.(ELC(1).GT.0.01))GO TO 707  BAS06350
C      ICOND=1                                                            BAS06360
C      IF(FLOAT(I2).LE.(FLOAT(N2)/10.))I2FLG=1                          BAS06370
C      707 CONTINUE                                                       BAS06380
C                                                                           BAS06390
C      C*** MOVE PHOTON TO NEXT EVENT POINT VIA BIASED TRAVERSE AND DETERMINE BAS06400
C      C*** POWER CONTRIBUTIONS                                           BAS06410
C                                                                           BAS06420
C      CALL TRAVRS(JTYPE,I2FLG,ICOND)                                     BAS06430
C      IF((NSCA.EQ.NSCA1).OR.(STH.LE.0.))GO TO 900                      BAS06440
C                                                                           BAS06450
C      C*** DETERMINE SCATTERING DIRECTION FOR NEXT TRAVERSE             BAS06460
C      C                                                                           BAS06470
C      CALL GAS(JTYPE)                                                    BAS06480
C                                                                           BAS06490
C      C*** DETERMINE BIASING DIRECTIONS FOR NEXT TRAVERSE               BAS06500
C      C                                                                           BAS06510
C      CALL ELM(XA,SA,EL)                                                 BAS06520
C      IF(EL(2).LE.EL(1))GO TO 900                                       BAS06530
C      GO TO 1700                                                         BAS06540
C      900 CONTINUE                                                       BAS06550
C                                                                           BAS06560
C      C*** END MULTIPLE SCATTERING LOOP, BEGIN CONVOLUTION BLOCK        BAS06570
C      C*** WRITE PARTIAL RUN RESULTS INTO STORAGE FILE FOR USE BY NEXT RUN BAS06580
C      C                                                                           BAS06590
C      DO 2201 NSCA=1,NSCAM                                               BAS06600
C      WRITE(KSTOR)ENC(NSCA)                                             BAS06610
C      DO 2201 II=1,NTMAX                                                 BAS06620
C      WRITE(KSTOR)ENC(NSCA,II)                                         BAS06630
C      2201 CONTINUE                                                       BAS06640
C      REWIND KSTOR                                                       BAS06650
C                                                                           BAS06660
C      C*** OUTPUT PARTIAL RUN RESULTS FOR STEADY STATE POWER            BAS06670
C      C                                                                           BAS06680
C      WRITE (IOOUT,7400)                                                 BAS06690
C      WRITE(IOOUT,7500)                                                 BAS06700
C      WRITE(IOOUT,7600)                                                 BAS06710
C      PTOT=0.                                                            BAS06720
C                                                                           BAS06730
C      C*****                                                             BAS06740
C      C      THE FOLLOWING STATEMENT SHOULD BE UNCOMMENTED IF OUTPUT TO   BAS06750
C      C      A USER-DEFINED PLOT FILE (NPLITU) IS DESIRED.              BAS06760
C      C      NSCAM= 1 + HIGHEST SIGNIFICANT ORDER OF SCATTERING         BAS06770
C      C      WRITE(NPLITU,9333) NSCAM                                     BAS06780
C      C      C*****                                                       BAS06790
C      C      DO 2110 NS=1,NSCAM                                           BAS06800
C      C      PE=0.                                                         BAS06810
C      C      NS1=NS-1                                                      BAS06820
C      C      DO 2115 NT=1,NTMAX                                           BAS06830
C      C      2115 PE=PE+ENC(NS,NT)                                         BAS06840
C      C      PE=PE/ENC(NS)                                                 BAS06850
C      C      PTOT=PTOT+PE                                                 BAS06860
C      C      C*****                                                       BAS06870
C      C      C      THE FOLLOWING STATEMENT SHOULD BE UNCOMMENTED IF OUTPUT TO   BAS06880
C      C      C      A USER-DEFINED PLOT FILE (NPLITU) IS DESIRED.              BAS06890
C      C      C      NS1 = ORDER OF SCATTERING                               BAS06900
C      C      C      PE = OBSERVED STEADY STATE POWER FOR THIS ORDER          BAS06910
C      C      C      ENC( )= TOTAL NUMBER OF PHOTONS COUNTED FOR THIS ORDER  BAS06920
C      C      C      WRITE(NPLITU,9333) NS1,PE,ENC(NS)                     BAS06930
C      C      C*****                                                       BAS06940
C      C      WRITE(IOOUT,7700)NS1,PE,ENC(NS)                             BAS06950
C      C      2110 CONTINUE                                                 BAS06960
C      C*****                                                             BAS06970
C      C      2110 CONTINUE                                                 BAS06980
C      C*****                                                             BAS06990
C      C*****                                                             BAS07000

```

```

C      THE FOLLOWING STATEMENTS SHOULD BE UNCOMMENTED IF OUTPUT TO
C      A USER-DEFINED PLOT FILE (NPLOTU) IS DESIRED.
C
C      PTOT = OBSERVED TOTAL STEADY STATE POWER (ALL ORDERS)
C9333 FORMAT(I2,1X,2(E9.4,1X))
C      WRITE(NPLOTU, 9222) PTOT
C*****
C      WRITE(IOOUT,7701)PTOT
C      DO 2500 NSCA=1,NSCAM
C
C**** DETERMINE INDEX OF TIME OF LAST NONZERO POWER VALUE FOR EACH
C**** ORDER
C
C      CALL SMOOZ(NSCA,NO)
C      NM(NSCA)=1+NO
C      IOR(NSCA)=NSCA-1
C2500 Y(NSCA)=NM(NSCA)
C
C**** DETERMINE LATEST TIME INDEX OF NONZERO POWER FOR ALL ORDERS
C
C      CALL GMAX(NSCAM,YMAX)
C      NMA=YMAX
C      JP=1
C2900 JP=JP+1
C      AT=TPUC(JP)
C      IF (AT.LE.0.0) GO TO 3000
C      GO TO 2900
C3000 JPMAX=JP-1
C      WRITE (IOOUT,7400)
C      WRITE (IOOUT,7900) JPMAX
C
C**** BEGIN PULSE LOOP
C
C      DO 3800 JP=1,JPMAX
C      WRITE (IOOUT,7400)
C      TP=TPUC(JP)
C      AT=(NMA+4)*DELT
C      IF(TP.GT.AT)TP=AT
C      IF(TP.LE.DELT)TP=DELT+1.E-3
C      NP=1.001+TP/DELT
C      NMAX=NMA+NP
C      IF(NMAX.GT.NDIM)NMAX=NDIM
C      NP=NMAX-NMA
C      TP=(NP-1)*DELT
C      TMAX=(NMAX-1)*DELT
C      WRITE (IOOUT,8100) JP,TP
C      WRITE (IOOUT,8200) JP,TMAX
C      IF(TP.LE.0.)GO TO 3999
C      DO 3400 NSCA=1,NSCAM
C      NMS=NM(NSCA)
C      NMS1=NMS-1
C
C**** NORMALIZE RETURN POWER BY DIVIDING CUMULATIVE POWER BY CUMULATIVE
C**** NUMBER OF PHOTONS
C
C      DO 3200 N=1,NMS1
C      ENC(NSCA,N)=ENC(NSCA,N)/ENC(NSCA)
C3200
C
C**** PERFORM SQUARE SOURCE PULSE CONVOLUTION WITH PROBABILITIES
C**** PER UNIT TIME
C
C      CALL CONV(NP,NMS,NMAX,NSCA)
C3400 CONTINUE
C      DO 3500 N=1,NMAX
C      Y(N)=0.0
C      DO 3500 NSCA=1,NSCAM
C      Y(N)=Y(N)+ENC(NSCA,N)
C3500 XN=-DELT
C      DO 3600 N=1,NMAX

```

```

BAS07010
BAS07020
BAS07030
BAS07040
BAS07050
BAS07060
BAS07070
BAS07080
BAS07090
BAS07100
BAS07110
BAS07120
BAS07130
BAS07140
BAS07150
BAS07160
BAS07170
BAS07180
BAS07190
BAS07200
BAS07210
BAS07220
BAS07230
BAS07240
BAS07250
BAS07260
BAS07270
BAS07280
BAS07290
BAS07300
BAS07310
BAS07320
BAS07330
BAS07340
BAS07350
BAS07360
BAS07370
BAS07380
BAS07390
BAS07400
BAS07410
BAS07420
BAS07430
BAS07440
BAS07450
BAS07460
BAS07470
BAS07480
BAS07490
BAS07500
BAS07510
BAS07520
BAS07530
BAS07540
BAS07550
BAS07560
BAS07570
BAS07580
BAS07590
BAS07600
BAS07610
BAS07620
BAS07630
BAS07640
BAS07650
BAS07660
BAS07670
BAS07680
BAS07690
BAS07700

```

```

      XN=XN+DELT
3600  X(N)=XN
C***  OUTPUT TIME-DEPENDENT RESULTS
      WRITE (IOOUT,8300)
      WRITE (IOOUT,8400)
      WRITE (IOOUT,8500) (IOR(NS),NS=1,NSCAM)
      WRITE (IOOUT,8501)
C*****
C      THE FOLLOWING STATEMENTS SHOULD BE UNCOMMENTED IF OUTPUT TO
      A USER-DEFINED PLOT FILE (NPLOTU) IS DESIRED.
      NMAX = NUMBER OF TIME BOXES USED FOR TIME-DEPENDENT DATA
      JP   = INDEX OF INPUT PULSE
      NSCAM = 1 + HIGHEST SIGNIFICANT ORDER OF SCATTERING
      X( ) = ARRAY OF TIME VALUES FOR EACH TIME BOX
      Y( ) = ARRAY OF OBSERVED TIME-DEPENDENT TOTAL POWER
      EN(L,)= ARRAY OF OBSERVED TIME-DEPENDENT POWER FOR ORDER L+1
      IF (I1.LT.N1) GO TO 9777
      WRITE (NPLOTU,9444) NMAX,JP,NSCAM
C9444  FORMAT(3(I3,1X))
      WRITE (NPLOTU,9222) (X(LLL),LLL=1,NMAX)
      WRITE (NPLOTU,9222) (Y(LLL),LLL=1,NMAX)
      DO 9666 LLL=1,NSCAM
      WRITE (NPLOTU,9222) (EN(LLL,LLX),LLX=1,NMAX)
C9666  CONTINUE
C9777  CONTINUE
C*****
      DO 3700 N=1,NMAX
      WRITE (IOOUT,8600) X(N),Y(N),EN(NSCA,N),NSCA=1,NSCAM)
3700  CONTINUE
C***  RELOAD PARTIAL RUN RESULTS FOR CONVOLUTION WITH NEXT PULSE
      DO 3701 NS=1,NSCAM
      READ (KSTOR) EN(NS)
      DO 3701 II=1,NTMAX
      READ (KSTOR) EN(NS,II)
3701  CONTINUE
      REWIND KSTOR
3800  CONTINUE
C***  END PULSE LOOP
1000  CONTINUE
C***  END OUTER PHOTON LOOP?
777  RETURN
C***  ERROR RETURN MESSAGES
3999  CONTINUE
      WRITE (IOOUT,8700) NTMAX
      RETURN
4000  CONTINUE
      WRITE (IOOUT,8800)
      RETURN
4001  CONTINUE
      WRITE (IOOUT,9000)
      RETURN
4002  CONTINUE
      WRITE (IOOUT,9001)
      RETURN
4003  CONTINUE
      WRITE (IOOUT,9002)
      RETURN
4004  CONTINUE

```

```

BAS07710
BAS07720
BAS07730
BAS07740
BAS07750
BAS07760
BAS07770
BAS07780
BAS07790
BAS07800
BAS07810
BAS07820
BAS07830
BAS07840
BAS07850
BAS07860
BAS07870
BAS07880
BAS07890
BAS07900
BAS07910
BAS07920
BAS07930
BAS07940
BAS07950
BAS07960
BAS07970
BAS07980
BAS07990
BAS08000
BAS08010
BAS08020
BAS08030
BAS08040
BAS08050
BAS08060
BAS08070
BAS08080
BAS08090
BAS08100
BAS08110
BAS08120
BAS08130
BAS08140
BAS08150
BAS08160
BAS08170
BAS08180
BAS08190
BAS08200
BAS08210
BAS08220
BAS08230
BAS08240
BAS08250
BAS08260
BAS08270
BAS08280
BAS08290
BAS08300
BAS08310
BAS08320
BAS08330
BAS08340
BAS08350
BAS08360
BAS08370
BAS08380
BAS08390
BAS08400

```

```

WRITE(1000,9005)
RETURN
4005 CONTINUE
WRITE(1000,9006)
RETURN
C*****FORMATS
4200 FORMAT (7F10.5)
4400 FORMAT (7I10)
4500 FORMAT (I10,2F10.5,I10)
4600 FORMAT (5E15.10)
4700 FORMAT (1H,34X,51(1H*))
4800 FORMAT (1H,34X,1H*,49X,1H*)
4900 FORMAT (1H,34X,1H*,9X,31HMONTE CARLO MULTIPLE SCATTERING,9X,1H*)
5000 FORMAT (1H,34X,1H*,16X,18HAEROSOL SCATTERING,15X,1H*)
5100 FORMAT (1H0,48X,23HPARAMETERS FOR THIS RUN)
5200 FORMAT (1H0,46X,28HUSER SUPPLIED PHASE FUNCTION)
5201 FORMAT (1H0,43X,34HMARITIME ARCTIC, VIS=0.1 TO 2.0 KM)
5202 FORMAT (1H0,47X,26HMARITIME POLAR, VIS=0.2 KM)
5203 FORMAT (1H0,47X,26HMARITIME POLAR, VIS=2.0 KM)
5204 FORMAT (1H0,42X,36HCONTINENTAL POLAR, VIS=0.2 TO 2.5 KM)
5205 FORMAT (1H0,52X,16HWHITE PHOSPHORUS)
5206 FORMAT (1H0,52X,16HHEXACHLOROETHANE)
5207 FORMAT (1H0,57X,7HFOG OIL)
5208 FORMAT (1H0,45X,31HDUST (MODERATE AEROSOL LOADING))
5209 FORMAT (1H0,46X,28HDUST (HEAVY AEROSOL LOADING))
5210 FORMAT (1H0,43X,34HMARITIME MODEL B, VIS=5 KM, RH=95%)
5211 FORMAT (1H0,43X,35HMARITIME MODEL B, VIS=10 KM, RH=90%)
5212 FORMAT (1H0,43X,35HMARITIME MODEL B, VIS=50 KM, RH=50%)
5500 FORMAT (1H0,46X,27HUSER SUPPLIED AEROSOL MODEL)
5600 FORMAT (1H,36X,11HWAVELENGTH=F6.3,16H MICROMETERS
1 7HALBEDO=F5.3)
5601 FORMAT (1H0,47X,25HELLIPSOIDAL AEROSOL CLOUD)
5602 FORMAT (1H,41X,36HCOORDINATE ORIGIN AT CENTER OF CLOUD)
5603 FORMAT (1H,38X,42HZ-AXIS VERTICAL, X-AXIS EAST, Y-AXIS NORTH)
5700 FORMAT (1H,36X,31HAEROSOL EXTINCTION COEFFICIENT=,
1 E10.4,7H KM**-1)
5701 FORMAT (1H0,51X,17HSOURCE PARAMETERS)
5702 FORMAT (1H,36X,27HSOURCE XYZ COORDINATES(KM)=,3(F8.4,1X))
5703 FORMAT (1H,36X,27HSOURCE APERTURE RADIUS(MM)=,F7.3)
5704 FORMAT (1H,36X,26HSOURCE AXIS POLAR ANGLE =,F7.3,8H DEGREES)
5705 FORMAT (1H,36X,26HSOURCE AXIS AZIMUTH ANGLE=,F7.3,8H DEGREES)
5706 FORMAT (1H,36X,26HSOURCE BEAM SPREAD ANGLE =,E10.4,8H RADIANS)
5900 FORMAT (1H0,50X,19HDETECTOR PARAMETERS)
6000 FORMAT (1H,35X,29H CONE OF VIEW HALF-ANGLE =,F7.3,
1 8H DEGREES)
6100 FORMAT (1H,35X,29H DETECTOR APERTURE RADIUS =,F7.3,
1 3H CM)
6201 FORMAT (1H,36X,29HDETECTOR XYZ COORDINATES(KM)=,3(F8.4,1X))
6400 FORMAT (1H,36X,28HDETECTOR AXIS POLAR ANGLE =,F7.3,
1 8H DEGREES)
6500 FORMAT (1H,36X,28HDETECTOR AXIS AZIMUTH ANGLE=,F7.3,
1 8H DEGREES)
6501 FORMAT (1H0,47X,23HGROUND PLANE PARAMETERS,/,40X,38HISOTROPIC REFLE
*CTION FROM GROUND PLANE)
6502 FORMAT (1H,36X,33HGROUND PLANE Z-COORDINATE ZG(KM)=,F7.3,/,
*37X,33HGROUND PLANE ALBEDO, ALBG, =,F7.3)
6800 FORMAT (1H0,34X,51(1H*))
6900 FORMAT (1H0,51X,16HCLLOUD PARAMETERS)
7000 FORMAT (1H,31X,43HELLIPSOID PRINCIPAL XYZ HALF-AXES(KM) =,
*3(F8.4,1X))
7100 FORMAT (1H,31X,43HEULER ANGLES THE,PHE,PSE OF ELLIPSOID( DEG)=,
*3(F8.4,1X))
7200 FORMAT (1H,31X,43HOPTICAL DEPTHS ALONG ELLIPSOID XYZ AXES =,
*3(F8.4,1X))
7400 FORMAT (1H0,100X)
7500 FORMAT (1H0,32X,53HSTEADY STATE POWER TO DETECTOR, FOR UNIT SOURCE
* POWER)
7600 FORMAT (1H0,37X,5HORDER,3X,10HSTEADY STATE POWER,3X,
1 17HNUMBER OF PHOTONS)
7700 FORMAT (1H,39X,12,8X,E10.5,8X,E12.6)

```

```

BAS0841
BAS0842
BAS0843
BAS0844
BAS0845
BAS0846
BAS0847
BAS0848
BAS0849
BAS0850
BAS0851
BAS0852
BAS0853
BAS0854
BAS0855
BAS0856
BAS0857
BAS0858
BAS0859
BAS0860
BAS0861
BAS0862
BAS0863
BAS0864
BAS0865
BAS0866
BAS0867
BAS0868
BAS0869
BAS0870
BAS0871
BAS0872
BAS0873
BAS0874
BAS0875
BAS0876
BAS0877
BAS0878
BAS0879
BAS0880
BAS0881
BAS0882
BAS0883
BAS0884
BAS0885
BAS0886
BAS0887
BAS0888
BAS0889
BAS0890
BAS0891
BAS0892
BAS0893
BAS0894
BAS0895
BAS0896
BAS0897
BAS0898
BAS0899
BAS0900
BAS0901
BAS0902
BAS0903
BAS0904
BAS0905
BAS0906
BAS0907
BAS0908
BAS0909
BAS0910

```

```

7701 FORMAT (1H,37X,5HTOTAL,7X,E10.5) BAS09110
7900 FORMAT (1H0,32X,23HPower INTO DETECTOR FOR,12,6H PULSE BAS09120
1 23H(S) OF DIFFERENT LENGTH) BAS09130
8100 FORMAT (1H0,37X,12HPULSE NUMBER,12,12H HAS LENGTH , BAS09140
1 E10.4,13H MICROSECONDS) BAS09150
8200 FORMAT (1H,23X,34HDETECTOR RESPONSE CUTOFF TIME FOR BAS09160
1 12HPULSE NUMBER,12,4H IS ,E10.4, BAS09170
2 13H MICROSECONDS) BAS09180
8300 FORMAT (1H0,25X,68HDETECTOR RESPONSE, POWER AS A FUNCTION OF TIME, BAS09190
* FOR UNIT PULSE POWER) BAS09200
8400 FORMAT (1H0,55X,21HPower FROM EACH ORDER/14X,5HTOTAL) BAS09210
8500 FORMAT (1H,3X,4HTIME,6X,5HPower,2X,10(X,12,4X)) BAS09220
8501 FORMAT(1H,130(1H-)) BAS09230
8600 FORMAT (12(E10.4,1X)) BAS09240
8700 FORMAT(1H0,6HNTMAX=,13,80H SHOULD BE DECREASED TO 46. IT IS TOO LABAS09250
*RGE TO ALLOW CONVOLUTION WITH YOUR PULSE) BAS09260
8800 FORMAT(1H0,97HYOUR INCIDENT PHOTONS NEVER INTERSECT THE CLOUD OR TBAS09270
*HE GROUND. CHECK YOUR INPUT SOURCE PARAMETERS) BAS09280
9000 FORMAT(1H0,72HTHE DETECTOR LOOKS ABOVE THE CLOUD. CHECK YOUR INPUTBAS09290
* DETECTOR PARAMETERS) BAS09300
9001 FORMAT(1H0,65HYOUR GROUND PLANE IS ENTIRELY ABOVE YOUR CLOUD. CHECBAS09310
*K YOUR INPUTS) BAS09320
9002 FORMAT(1H0,76HNEITHER YOUR SOURCE NOR YOUR DETECTOR LOOK INTO THE BAS09330
*CLOUD. CHECK YOUR INPUTS) BAS09340
9003 FORMAT(1H0,71HYOUR SOURCE WAS UNDERGROUND. IT HAS BEEN PUT AT THE BAS09350
*GROUND, WITH XS(3)=,F6.3,2HKM) BAS09360
9004 FORMAT(1H0,114HYOUR GROUND PLANE WAS TOO FAR AWAY FROM THE CLOUD TBAS09370
*O PRODUCE GROUND REFLECTIONS WITHIN THE MAX TIME DELAY ALLOWED.,/, BAS09380
*34H THE GROUND PLANE WAS MOVED TO ZG=,F6.3,2HKM) BAS09390
9005 FORMAT(1H0,91HGROUND PLANE WAS ABSENT AND DETECTOR DOES NOT LOOK ABAS09400
*T CLOUD. CHECK YOUR DETECTOR PARAMETERS) BAS09410
9006 FORMAT(1H0,79HGROUND PLANE WAS ABSENT AND SOURCE DOES NOT ILLUMINABAS09420
*TE CLOUD. CHECK YOUR INPUTS) BAS09430
END BAS09440

```

```

SUBROUTINE BKWD(JTYPE)
COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK
COMMON /RNDM/ SEED
COMMON /MOD/V1,C1,S1,SA(3),EN(10,100),ENC(10),ELMIN,DELD,DTOT,
+NTMAX,NSCAM,KMAX,LMAX,LMM1
COMMON /BASPOT/U(65),SUM(65),WVL(16),NWVL,ALBED(16),BS(16),
+BE(16),SINGWV,PF(65),LLMAX
COMMON /FAHT/SD(3),UV,GAMMA,ELD(2),STH,FAC,ALIM,THV,TAU,EL(2),
+ALB(2),ZG,DMAX
COMMON /FWD/AKM,R(3,3),AKSQ,XD(3),ASQ(3),RE(3,3),A(3)
COMMON /FGEL/XA(3),D,NSCA
DIMENSION SBC(3),SBA(3)
NSCAP=NSCA+1

C
C*** DETERMINE MONTE CARLO BACKWARD TRAVERSE ANGLES THETA (AT)
C*** AND PHI (BT) INSIDE DETECTOR CONE.
C
BT=TWOPI*RAND ( SEED )
AT=THV*RAND ( SEED )
IF(AT.LE.1.E-30) AT=0.
IF(BT.LE.1.E-30) BT=0.
V=SIN(AT)
U23=COS(AT)

C
C*** ROTATE BACKWARD TRAVERSE VECTOR INTO STANDARD FRAME OF REFERENCE.
C
CALL ROTAT(AT,BT,R,SBC)

C
C*** DETERMINE BIASING DISTANCES FOR BACKWARD TRAVERSE.
C
CALL ELM(XD,SBC,EL)

C
C*** DETERMINE STATISTICAL WEIGHT REX OF BACKWARD TRAVERSE.
C
REX=1.-EXP(-GAMMA*(EL(2)-EL(1)))
REXRN=REX*RAND(SEED)
IF((1.-REXRN).LE.1.E-7) GO TO 15

C
C*** DETERMINE RANDOM DISTANCE FOR BIASED BACKWARD TRAVERSE.
C
ELBC=-ALOG(1.-REXRN)/GAMMA+EL(1)
CT=0.
DO 4 K=1,3
SBA(K)=XD(K)-XA(K)+SBC(K)*ELBC
4 CT=CT+SBA(K)**2
IF(CT.LE.1.E-30) CT=0.
ELAB=SQRT(CT)
IF(ELAB.LT.ALIM)GO TO 15

C
C*** DETERMINE TIME BOX INDEX NT FOR THE COMPLETE BACKWARD TRAVERSE.
C
NT=1.+(DTOT+ELAB+ELBC-ELMIN)/DELD
IF(NT.LT.0) GO TO 15
IF(NT.EQ.0) NT=1
IF(NT.GT.NTMAX)RETURN
U1=0.
U2=0.
DO 5 K=1,3
SBA(K)=SBA(K)/ELAB
U1=U1+SBA(K)*SBA(K)
5 U2=U2-SBA(K)*SBC(K)
6 CALL FIND(U2,PF2)
IF(JTYPE.EQ.2)GO TO 9
CALL FIND(U1,PF1)
GO TO 10
9 CONTINUE
CALL ELM(XA,SBA,EL)
ELAB=ELAB-EL(1)
PF1=1.
10 DOM=FAC*ALB(1)*STH*REX*V*U23*PF1*PF2*EXP(-GAMMA*ELAB)/CT

```

```

BKW0001
BKW0002
BKW0003
BKW0004
BKW0005
BKW0006
BKW0007
BKW0008
BKW0009
BKW0010
BKW0011
BKW0012
BKW0013
BKW0014
BKW0015
BKW0016
BKW0017
BKW0018
BKW0019
BKW0020
BKW0021
BKW0022
BKW0023
BKW0024
BKW0025
BKW0026
BKW0027
BKW0028
BKW0029
BKW0030
BKW0031
BKW0032
BKW0033
BKW0034
BKW0035
BKW0036
BKW0037
BKW0038
BKW0039
BKW0040
BKW0041
BKW0042
BKW0043
BKW0044
BKW0045
BKW0046
BKW0047
BKW0048
BKW0049
BKW0050
BKW0051
BKW0052
BKW0053
BKW0054
BKW0055
BKW0056
BKW0057
BKW0058
BKW0059
BKW0060
BKW0061
BKW0062
BKW0063
BKW0064
BKW0065
BKW0066
BKW0067
BKW0068
BKW0069
BKW0070

```

ENC(NSCAP, NT)=ENC(NSCAP, NT)+DOM
RETURN
15 ENC(NSCAP)=ENC(NSCAP)-1.
STH=0,
RETURN
END

BKW00710
BKW00720
BKW00730
BKW00740
BKW00750
BKW00760

SUBROUTINE CONV(NP,NMS,NMAX,NSCA)	CON00010
C*****THIS SUBROUTINE CONVOLUTES A GIVEN SEQUENCE OF VALUES EN(NSCA,N),	CON00020
C N=1,NMS, WITH THE UNIT SQUARE FUNCTION EXTENDING FROM TIME=0 TO	CON00030
C*****TIME=(NP-1)*DELT.	CON00040
DOUBLE PRECISION DBLE,XDBLE(100)	CON00050
COMMON/CONB/X(100),Y(100)	CON00060
COMMON /MO5/V1,C1,S1,SA(3),EN(10,100),ENC(10),ELMIN,DELD,DTOT,	CON00070
+NTMAX,NSCAN,KMAX,LMAX,LMM1	CON00080
COMMON /BASPOT/U(65),SUM(65),WVL(16),NWVL,ALBED(16),BS(16),	CON00090
+BE(16),SINGWV,PF(65),LLMAX	CON00100
NMS1=NMS+1	CON00110
NMA=NMS+NP-1	CON00120
XDBLE(1)=0.E-00	CON00130
DO 100 N=2,NMS	CON00140
100 XDBLE(N)=XDBLE(N-1)+DBLE(EN(NSCA,N-1))	CON00150
NP1=NP+1	CON00160
IF (NP.LT.NMS) GO TO 600	CON00170
DO 200 N=1,NMS	CON00180
200 Y(N)=XDBLE(N)	CON00190
IF (NP.EQ.NMS) GO TO 400	CON00200
DO 300 N=NMS1,NP	CON00210
300 Y(N)=XDBLE(NMS)	CON00220
400 CONTINUE	CON00230
DO 500 N=NP1,NMA	CON00240
500 Y(N)=XDBLE(NMS)-XDBLE(N-NP+1)	CON00250
GO TO 1000	CON00260
600 CONTINUE	CON00270
DO 700 N=1,NP	CON00280
700 Y(N)=XDBLE(N)	CON00290
DO 800 N=NP1,NMS	CON00300
800 Y(N)=XDBLE(N)-XDBLE(N-NP+1)	CON00310
DO 900 N=NMS1,NMA	CON00320
900 Y(N)=XDBLE(NMS)-XDBLE(N-NP+1)	CON00330
1000 CONTINUE	CON00340
IF (NMA.EQ.NMAX) GO TO 1200	CON00350
NMA1=NMA+1	CON00360
DO 1100 N=NMA1,NMAX	CON00370
1100 Y(N)=0.0	CON00380
DO 1150 N=1,NMAX	CON00390
1150 EN(NSCA,N)=Y(N)	CON00400
1200 RETURN	CON00410
END	CON00420

C	SUBROUTINE ELM(X1,S1,EL)	ELM00010
C***	THIS SUBROUTINE DETERMINES THE SMALLEST (EL(1)) AND THE LARGEST	ELM00020
C***	(EL(2)) DISTANCES FROM A POINT X1() TO THE SURFACES OF AN	ELM00030
C***	ELLIPSOID ALONG A LINE OF SIGHT DEFINED BY UNIT VECTOR S1().	ELM00040
C		ELM00050
	COMMON/FWD/AKM,R(3,3),AKSQ,XD(3),ASQ(3),RE(3,3),A(3)	ELM00060
	DIMENSION X1(3),S1(3),EL(2)	ELM00070
	AT=0.	ELM00080
	BT=0.	ELM00090
	CT=-1.	ELM00100
	DO 1 K=1,3	ELM00110
	X=0.	ELM00120
	Y=0.	ELM00130
	DO 2 L=1,3	ELM00140
	X=X+RE(K,L)*X1(L)	ELM00150
2	Y=Y+RE(K,L)*S1(L)	ELM00160
	AT=AT+Y**2/ASQ(K)	ELM00170
	BT=BT+X*Y/ASQ(K)	ELM00180
1	CT=CT+X**2/ASQ(K)	ELM00190
	DISC=BT**2-AT*CT	ELM00200
	IF(DISC.LT.1.E-30)GO TO 10	ELM00210
	DISC=SQRT(DISC)	ELM00220
	EL(1)=-BT+DISC)/AT	ELM00230
	EL(2)=(-BT-DISC)/AT	ELM00240
	IF(CT.LE.0.)EL(1)=0.	ELM00250
	IF((EL(1).LT.0.).OR.(EL(2).LT.0.))GO TO 10	ELM00260
	RETURN	ELM00270
10	EL(1)=0.	ELM00280
	EL(2)=0.	ELM00290
	RETURN	ELM00300
	END	ELM00310
		ELM00320

```

SUBROUTINE FIND(U1,PFU)
COMMON /MOS/V1,C1,S1,SA(3),ENC(10,100),ENC(10),ELMIN,DELD,DTOT,
+NTMAX,NSCAM,KMAX,LMAX,LMM1
COMMON /BASPOT/U(65),SUM(65),WVL(16),NWVL,ALBED(16),BS(16),
+BE(16),SINGWV,PF(65),LLMAX
L=1
LL=LMM1
DO 1 K=1,KMAX
LL=LL/2
L=L+LL
AT=U1-U(L)
IF(AT.GT.1.E-7)L=L-LL
1 CONTINUE
PFU=PF(L)+(U1-U(L))*(PF(L+1)-PF(L))/(U(L+1)-U(L))
RETURN
END

```

```

FIN00010
FIN00020
FIN00030
FIN00040
FIN00050
FIN00060
FIN00070
FIN00080
FIN00090
FIN00100
FIN00110
FIN00120
FIN00130
FIN00140
FIN00150
FIN00160

```

SUBROUTINE FWRD(SCA,DOM,ELAC)	FWR00010
COMMON/CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK	FWR00020
COMMON/FGEL/XA(3),D,NSCA	FWR00030
COMMON/FWD/AKM,R(3,3),AKSQ,XD(3),ASQ(3),RE(3,3),A(3)	FWR00040
COMMON/RNDM/ SEED	FWR00050
DIMENSION SCA(3)	FWR00060
PSI=TWOPI*RAND (SEED)	FWR00070
TH=PI2	FWR00080
CALL ROTAT(TH,PSI,R,SCA)	FWR00090
ANR=RND(SEED)	FWR00100
IF(ANR.LE.1.E-30) ANR=0.	FWR00110
RHO=AKM*SQRT(ANR)	FWR00120
ELSQ=0.	FWR00130
DO 1 K=1,3	FWR00140
SCA(K)=XD(K)-XA(K)+SCA(K)*RHO	FWR00150
1 ELSQ=ELSQ+SCA(K)**2	FWR00160
IF(ELSQ.LE.1.E-30) ELSQ=1.E-30	FWR00170
ELAC=SQRT(ELSQ)	FWR00180
DO 3 K=1,3	FWR00190
3 SCA(K)=SCA(K)/ELAC	FWR00200
DOM=AKSQ/(4.*ELSQ)	FWR00210
RETURN	FWR00220
END	FWR00230

```

SUBROUTINE GAS(JTYPE)
C
C*** THIS SUBROUTINE DETERMINES A RANDOMLY-SELECTED SCATTERING
C*** DIRECTION USED IN PHOTON TRAVERSES WITHIN THE ELLIPSOIDAL
C*** AEROSOL CLOUD. JTYPE=1 SIGNIFIES THAT THE SCATTERING EVENT AT
C*** WHICH THIS ANGLE IS SELECTED IS WITHIN THE AEROSOL CLOUD.
C*** JTYPE=2 SIGNIFIES THAT THE SCATTERING EVENT IS ON THE GROUND
C*** PLANE.
C
COMMON/ALL/AT,BT,CT,BLIM
COMMON/CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK
COMMON /MO5/V1,C1,S1,SA(3),EN(10,100),ENC(10),ELMIN,DELD,DTOT,
+NTMAX,NSCAM,KMAX,LMAX,LMM1
COMMON /BASPOT/U(65),SUM(65),WVL(16),NWVL,ALBED(16),BS(16),
+BE(16),SINGWV,PF(65),LLMAX
COMMON/RNDM/ SEED
COMMON/FGEL/XA(3),D,NSCA
AT=Rand ( SEED )
IF(AT.LE.1.E-30) AT=0.
IF(JTYPE.EQ.1)GO TO 1
SA(3)=1.-AT
ARG=1.-SA(3)**2
IF(ARG.LT.1.E-30) ARG=0.
BT=TWOPI*Rand ( SEED )
IF(BT.LE.1.E-30) BT=0.
V1=SQRT(ARG)
C1=cos(BT)
S1=sin(BT)
SA(1)=V1*C1
SA(2)=V1*S1
RETURN
1 CALL USCA(AT,BT)
CT=TWOPI*Rand ( SEED )
IF(CT.LE.1.E-30) CT=0.
ARG=1.-BT**2
IF(ARG.LT.1.E-30) ARG=0.
V=SQRT(ARG)
C=cos(CT)
S=sin(CT)
SA(1)=BT*SA(1)+V*(C*C1*SA(3)-S*S1)
SA(2)=BT*SA(2)+V*(C*S1*SA(3)+S*C1)
SA(3)=BT*SA(3)-V*C*V1
AT=SA(1)**2+SA(2)**2
IF(AT.LT.1.E-10)GO TO 3
V1=SQRT(AT)
C1=SA(1)/V1
S1=SA(2)/V1
RETURN
3 C1=C
S1=S
V1=0.
SA(1)=0.
SA(2)=0.
BT=SA(3)
SA(3)=1.
IF(BT.LT.0.)SA(3)=-1.
RETURN
END

```

```

GAS00010
GAS00020
GAS00030
GAS00040
GAS00050
GAS00060
GAS00070
GAS00080
GAS00090
GAS00100
GAS00110
GAS00120
GAS00130
GAS00140
GAS00150
GAS00160
GAS00170
GAS00180
GAS00190
GAS00200
GAS00210
GAS00220
GAS00230
GAS00240
GAS00250
GAS00260
GAS00270
GAS00280
GAS00290
GAS00300
GAS00310
GAS00320
GAS00330
GAS00340
GAS00350
GAS00360
GAS00370
GAS00380
GAS00390
GAS00400
GAS00410
GAS00420
GAS00430
GAS00440
GAS00450
GAS00460
GAS00470
GAS00480
GAS00490
GAS00500
GAS00510
GAS00520
GAS00530
GAS00540
GAS00550
GAS00560
GAS00570
GAS00580

```

	SUBROUTINE GMAX(IMAX,YMAX)	GMA00010
C		GMA00020
C***	THIS SUBROUTINE DETERMINES THE MAXIMUM VALUE YMAX OF AN INPUT	GMA00030
C***	ARRAY Y() OF DIMENSION IMAX.	GMA00040
C		GMA00050
	COMMON/CONB/X(100),Y(100)	GMA00060
	I=0	GMA00070
	YMAX=Y(1)	GMA00080
I	I=I+1	GMA00090
	IF(I.EQ.IMAX)RETURN	GMA00100
	T=Y(I+1)-YMAX	GMA00110
	IF(T.GT.0.)YMAX=Y(I+1)	GMA00120
	GO TO 1	GMA00130
	END	GMA00140

C	SUBROUTINE MATRX(TH,PH,S,R)	MAT00010
C***	THIS SUBROUTINE GENERATES UNIT VECTOR S() AND ROTATION MATRIX	MAT00020
C***	R() FOR A SET OF INPUT POLAR ANGLES (TH,PH).	MAT00030
C***	THE ROTATION MATRIX R() ROTATES A VECTOR DEFINED RELATIVE TO THE	MAT00040
C***	(TH,PH) DIRECTION INTO THE STANDARD SYSTEM OF COORDINATES.	MAT00050
C***	THE UNIT VECTOR S() POINTS IN THE (TH,PH) DIRECTION IN THE	MAT00060
C***	STANDARD SYSTEM OF COORDINATES.	MAT00070
C		MAT00080
	COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK	MAT00090
	DIMENSION S(3),R(3,3)	MAT00100
	AT=PIRAD*TH	MAT00110
	BT=PIRAD*PH	MAT00120
	IF(AT.LE.1.E-30) AT=0.	MAT00130
	IF(BT.LE.1.E-30) BT=0.	MAT00140
	V1=SIN(AT)	MAT00150
	C1=COS(AT)	MAT00160
	S1=SIN(BT)	MAT00170
	S(1)=V1*C1	MAT00180
	S(2)=V1*S1	MAT00190
	S(3)=COS(AT)	MAT00200
	R(1,1)=C1*S(3)	MAT00210
	R(1,2)=S1*S(3)	MAT00220
	R(1,3)=-V1	MAT00230
	R(2,1)=-S1	MAT00240
	R(2,2)=C1	MAT00250
	R(2,3)=0.	MAT00260
	DO 1 K=1,3	MAT00270
1	R(3,K)=S(K)	MAT00280
	RETURN	MAT00290
	END	MAT00300
		MAT00310

```

SUBROUTINE ROTAT(TH,PH,R,S)
DIMENSION R(3,3),X(3),S(3)
IF(TH.LE.1.E-30) TH=0.
IF(PH.LE.1.E-30) PH=0.
V=SIN(TH)
X(1)=V*COS(PH)
X(2)=V*SIN(PH)
X(3)=COS(TH)
DO 1 J=1,3
S(J)=0.
DO 1 K=1,3
1 S(J)=S(J)+R(K,J)*X(K)
RETURN
END

```

```

ROT00010
ROT00020
ROT00030
ROT00040
ROT00050
ROT00060
ROT00070
ROT00080
ROT00090
ROT00100
ROT00110
ROT00120
ROT00130
ROT00140

```


SUBROUTINE SMOOZ(NSCA,NO)	SMO00010
C*****THIS SUBROUTINE DETERMINES 'NO', THAT VALUE OF I BEYOND WHICH ALL	SMO00020
C*****Y(I) IN A SEQUENCE ARE ZERO.	SMO00030
COMMON /MO5/YI,C1,S1,SA(3),EN(10,100),ENC(10),ELMIN,DELD,DTOT,	SMO00040
+NTMAX,NSCAM,KMAX,LMAX,LMM1	SMO00050
COMMON /BASPOT/UK(65),SUM(65),WVL(16),NWVL,ALBED(16),BS(16),	SMO00060
+BE(16),SINGWV,PF(65),LLMAX	SMO00070
I=NTMAX+1	SMO00080
100 I=I-1	SMO00090
IF(I.EQ.0)GO TO 200	SMO00100
YI=ENC(NSCA,I)	SMO00110
IF (YI.GT.0.0) GO TO 200	SMO00120
GO TO 100	SMO00130
200 NO=I	SMO00140
RETURN	SMO00150
END	SMO00160

```

C      SUBROUTINE START(SS)
C***  THIS SUBROUTINE INITIALIZES PHOTON LAUNCH DIRECTION AND CALCULATES
C***  DIRECT BEAM (ZEROth ORDER) CONTRIBUTIONS TO RECEIVED POWER.
C      COMMON/ALL/AT,BT,CT,BLIM
C      COMMON/CONB/X(100),Y(100)
C      COMMON/FWD/AKM,R(3,3),AKSQ,XD(3),ASQ(3),RE(3,3),A(3)
C      COMMON/HIT/UDS,THSP,RS(3,3),XS(3),DSA,XV(3)
C      COMMON/FGEL/XA(3),D,NSCA
C      COMMON/FAHT/SD(3),UV,GAMMA,ELD(2),STH,FAC,ALIM,THV,TAU,EL(2),
C      *ALB(2),ZG,DMAX
C      COMMON/RNDM/ SEED
C      COMMON /MOS/V1,C1,S1,SA(3),EN(10,100),ENC(10),ELMIN,DELD,DTOT,
C      *NTMAX,NSCAM,KMAX,LMAX,LMM1
C      COMMON /BASPOT/U(65),SUM(65),WVL(16),NWVL,ALBED(16),BS(16),
C      *BE(16),SINGWV,PF(65),LLMAX
C      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK
C      DIMENSION SS(3)
C***  GENERATE RANDOM THETA (AT) AND PHI (BT) PHOTON LAUNCH ANGLES
C***  CONFINED WITHIN SOURCE CONE.
C      ANR=RNDM(SEED)
C      IF(ANR.LE.1.E-30) ANR=0.
C      AT=THSP*SQRT(ANR)
C      BT=TWOPI*RNDM ( SEED)
C***  ROTATE LAUNCH VECTOR SA( ) INTO STANDARD FRAME OF REFERENCE.
C      CALL ROTAT(AT,BT,RS,SA)
C***  DETERMINE HORIZONTAL COMPONENT CT OF LAUNCH VECTOR AND BRANCH
C***  TO VERTICAL TREATMENT (WHICH INCLUDES GROUND PLANE) IF THIS
C***  COMPONENT IS VERY SMALL.
C      CT=SA(1)**2+SA(2)**2
C      IF(CT.LE.1.E-24)GO TO 1
C      V1=SQRT(CT)
C      C1=SA(1)/V1
C      S1=SA(2)/V1
C      GO TO 2
C***  DEFINE VERTICAL UNIT VECTOR
C      1 V1=0.
C      CT=SA(3)
C      SA(3)=1.
C      IF((CT.LT.0.).AND.(ALB(2).GT.0.)) SA(3)=-1.
C      C1=1.
C      S1=0.
C      SA(1)=0.
C      SA(2)=0.
C***  INITIALIZE TOTAL TRAVERSE DISTANCE DTOT AND STATISTICAL
C***  STRENGTH STH.
C      2 DTOT=0.
C      STH=1.
C      BT=0.
C      IF(AT.LE.1.E-30) GO TO 11
C      DAA=DSA/COS(AT)
C      DAD=0.
C      DO 3 K=1,3
C      XA(K)=XV(K)+DAA*SA(K)
C      DAD=DAD+Y(K)**2
C      3 BT=BT+Y(K)*SS(K)
C      IF(DAD.LE.1.E-30) DAD=0.
C      DAD=SQRT(DAD)
C
C      STA00010
C      STA00020
C      STA00030
C      STA00040
C      STA00050
C      STA00060
C      STA00070
C      STA00080
C      STA00090
C      STA00100
C      STA00110
C      STA00120
C      STA00130
C      STA00140
C      STA00150
C      STA00160
C      STA00170
C      STA00180
C      STA00190
C      STA00200
C      STA00210
C      STA00220
C      STA00230
C      STA00240
C      STA00250
C      STA00260
C      STA00270
C      STA00280
C      STA00290
C      STA00300
C      STA00310
C      STA00320
C      STA00330
C      STA00340
C      STA00350
C      STA00360
C      STA00370
C      STA00380
C      STA00390
C      STA00400
C      STA00410
C      STA00420
C      STA00430
C      STA00440
C      STA00450
C      STA00460
C      STA00470
C      STA00480
C      STA00490
C      STA00500
C      STA00510
C      STA00520
C      STA00530
C      STA00540
C      STA00550
C      STA00560
C      STA00570
C      STA00580
C      STA00590
C      STA00600
C      STA00610
C      STA00620
C      STA00630
C      STA00640
C      STA00650
C      STA00660
C      STA00670
C      STA00680
C      STA00690
C      STA00700

```

C*** DETERMINE DIRECT BEAM ATTENUATION DISTANCE AND BIASING DISTANCES
 C*** FOR INITIAL PHOTON TRAVERSE.

C

```

    CALL ELM(XA,SA,EL)
    IF(BT.LE.ALIM) GO TO 4
    IF(DAD.LE.(1.001*DSA)) GO TO 4
    BT=0.
    CT=0.
    DO 5 K=1,3
    CT=CT+SA(K)*SD(K)
  5 BT=BT+Y(K)*SD(K)
    CTSQ=CT**2
    IF(CTSQ.LE.1.E-30) GO TO 4
    ELSD=ABS(BT/CT)-DAA
    CT=0.
    DO 6 K=1,3
    CT=CT+(XA(K)+SA(K)*ELSD-XD(K))**2
  6 IF(CT.GT.AKSQ) GO TO 4
    IF(EL(2).LE.EL(1)) STH=0.
    IF(ELD(1).LE.0.) EL(2)=ELSD
    IF(DAD.LT.ALIM) GO TO 4
    UDS=-BT/DAD
    IF(UDS.LT.UV) RETURN
    NT=1.+(ELSD-ELMIN)/DELD
    IF(NT.LT.0) GO TO 11
    IF(NT.EQ.0) NT=1
    EN(1,NT)=EN(1,NT)+EXP(-GAMMA*(EL(2)-EL(1)))
    RETURN
  4 CONTINUE
    IF((EL(2)-EL(1)).GT.1.E-20) RETURN
    IF((SA(3).GE.0.)OR.(ALB(2).LE.1.E-20)) STH=0.
    RETURN
  11 STH=0.
    RETURN
  END

```

STA00710
 STA00720
 STA00730
 STA00740
 STA00750
 STA00760
 STA00770
 STA00780
 STA00790
 STA00800
 STA00810
 STA00820
 STA00830
 STA00840
 STA00850
 STA00860
 STA00870
 STA00880
 STA00890
 STA00900
 STA00910
 STA00920
 STA00930
 STA00940
 STA00950
 STA00960
 STA00970
 STA00980
 STA00990
 STA01000
 STA01010
 STA01020
 STA01030
 STA01040
 STA01050

SUBROUTINE TRAVERS(JTYPE,I2FLG,ICOND)	TRA00010
COMMON/RNDM/ SEED	TRA00020
COMMON/ALL/AT,BT,CT,BLIM	TRA00030
COMMON/FAHT/SD(3),UV,GAMMA,ELD(2),STH,FAC,ALIM,THV,TAU,EL(2),	TRA00040
*ALB(2),ZG,DMAX	TRA00050
COMMON /M05/V1,C1,S1,SA(3),EN(10,100),ENC(10),ELMIN,DELD,DTOT,	TRA00060
+NTMAX,NSCAM,KMAX,LMAX,LMM1	TRA00070
COMMON /BASPT/U(65),SUN(65),WVL(16),NWVL,ALBED(16),BS(16),	TRA00080
+BE(16),SINGWV,PF(65),LLMAX	TRA00090
COMMON/FGEL/XA(3),D,NSCA	TRA00100
COMMON/FWD/AKM,R(3,3),AKSQ,XD(3),ASQ(3),RE(3,3),A(3)	TRA00110
COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK	TRA00120
COMMON/CONB/X(100),Y(100)	TRA00130
DIMENSION SDA(3)	TRA00140
DGAM=EL(2)-EL(1)	TRA00150
REXIT=.99999	TRA00160
RX=.99999	TRA00170
RXX=.99999	TRA00180
IF((DGAM.LE.0.).AND.(ALB(2).GT.0.))GO TO 9	TRA00190
IF(DGAM.LE.1.E-7) GO TO 70	TRA00200
IF(I2FLG.EQ.1) GO TO 5	TRA00210
IF((SA(3).GE.0.).OR.(ALB(2).LE.0.))REXIT=1.-EXP(-GAMMA*DGAM)	TRA00220
AT=EL(1)-(ALOG(1.-REXIT*RAND (SEED)))/GAMMA	TRA00230
GO TO 4	TRA00240
5 CONTINUE	TRA00250
IF(ELD(1).GT.ALIM)ALIM=ELD(1)	TRA00260
RXX=((1./GAMMA)*((1./ALIM)-(1./DGAM))	TRA00270
DENOM=((1./ALIM)-GAMMA*RXX*RAND(SEED))	TRA00280
IF(DENOM.LE.1.E-7) GO TO 70	TRA00290
AT=1./DENOM	TRA00300
RX=EXP(-(GAMMA*AT))*((AT*GAMMA)**2	TRA00310
BT=XA(3)+SA(3)*AT	TRA00320
IF((BT.GT.ZG).AND.(AT.LE.EL(2)))GO TO 2	TRA00330
9 JTYPE=2	TRA00340
AT=(ZG-XA(3))/SA(3)	TRA00350
GO TO 1	TRA00360
2 JTYPE=1	TRA00370
1 STH=ALB(JTYPE)*REXIT*STH*RXX	TRA00380
IF(STH.LE.0.)RETURN	TRA00390
DO 3 K=1,3	TRA00400
3 XA(K)=XA(K)+AT*SA(K)	TRA00410
DTOT=DTOT+AT	TRA00420
IF(NSCA.NE.2) GO TO 50	TRA00430
AT=0.	TRA00440
DO 8 K=1,3	TRA00450
8 SDA(K)=XD(K)-XA(K)	TRA00460
AT=AT+SDA(K)*SDA(K)	TRA00470
IF(AT.LT.BLIM)GO TO 15	TRA00480
AT=SQRT(AT)	TRA00490
BT=0.	TRA00500
DO 32 K=1,3	TRA00510
SDA(K)=SDA(K)/AT	TRA00520
32 BT=BT-SDA(K)*SD(K)	TRA00530
IF(BT.LT.UV) GO TO 50	TRA00540
CALL FWRD(SDA,DOM,AT)	TRA00550
NT=1.+(DTOT+AT-ELMIN)/DELD	TRA00560
IF(NT.LT.0) RETURN	TRA00570
IF(NT.EQ.0) NT=1	TRA00580
IF((NT.EQ.1).AND.(I2FLG.EQ.0).AND.(ICOND.EQ.1))GO TO 50	TRA00590
IF(NT.GT.NTMAX)RETURN	TRA00600
CT=0.	TRA00610
BT=0.	TRA00620
DO 7 K=1,3	TRA00630
BT=BT-SDA(K)*SD(K)	TRA00640
7 CT=CT+SDA(K)*SA(K)	TRA00650
PF1=1.	TRA00660
IF(JTYPE.EQ.1)CALL FIND(CT,PF1)	TRA00670
CALL ELM(XA,SDA,EL)	TRA00680
AT=AT-EL(1)	TRA00690
IF(ELD(1).GT.0.)AT=EL(2)-EL(1)	TRA00700

```

DOM=DOM*PFI*STH*EXP(-GAMMA*AT)*BT*RX
IF(I2FLG.NE.1) GO TO 13
IF(NT.EQ.1) DOM=DOM*10.
IF(NT.GT.1) DOM=0.
GO TO 14
13 CONTINUE
IF(ICOND.EQ.1) DOM=DOM/0.9
14 EN(NSCA,NT)=EN(NSCA,NT)+DOM
GO TO 50
15 ENC(2)=ENC(2)-1.
STH=0.
RETURN
50 CONTINUE
IF(I2FLG.EQ.1) GO TO 60
CALL BKWD(JTYPE)
60 RETURN
70 ENC(NSCA)=ENC(NSCA)-1.
STH=0.
RETURN
END

```

```

TRA00710
TRA00720
TRA00730
TRA00740
TRA00750
TRA00760
TRA00770
TRA00780
TRA00790
TRA00800
TRA00810
TRA00820
TRA00830
TRA00840
TRA00850
TRA00860
TRA00870
TRA00880
TRA00890
TRA00900

```

SUBROUTINE USCA(SC,US)	USC00010
COMMON /M05/V1,C1,S1,SA(3),ENC(10,100),ENC(10),ELMIN,DELD,DTOT,	USC00020
+NTMAX,NSCAM,KMAX,LMAX,LMM1	USC00030
COMMON /BASPOT/U(65),SUM(65),WVL(16),NWVL,ALBED(16),BS(16),	USC00040
+BE(16),SINGWV,PF(65),LLMAX	USC00050
L=1	USC00060
LL=LMM1	USC00070
DO 1 K=1,KMAX	USC00080
LL=LL/2	USC00090
L=L+LL	USC00100
IF(SUM(L).GT.SC)L=L-LL	USC00110
1 CONTINUE	USC00120
US=U(L)+(SC-SUM(L))*(U(L+1)-U(L))/(SUM(L+1)-SUM(L))	USC00130
RETURN	USC00140
END	USC00150

```

SUBROUTINE SMOKE(WAVE1,ICLMAT,TRANS,IERR)
COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUS
COMMON /CLYMAT/TEMP,PRESS,RH,AH,DP,VIS,CLDAMT,CLDHYT,
1 FOGPRB,WINDVEL,WINDDIR,IPASCT
COMMON /GEOMET/PTS(15),IGEOSW
COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDECK
DIMENSION IR(26),ITTL(11,5),TSUB(11),QDIV(11)
EXTERNAL JPASCT
SMK00010
SMK00020
SMK00030
SMK00040
SMK00050
SMK00060
SMK00070
SMK00080

C ***** NEW PROGRAM OPTIONS ADDED AS REVISIONS 03 & 04

C THIS REVISION TO THE MODEL ALLOWS THE OPTION OF PRODUCING A ONE
C DIMENSIONAL "SNAPSHOT" OF A SMOKE SCREEN DUE TO ANY NUMBER OF
C MUNITIONS AT SOME SPECIFIED TIME; OUTPUT FOR THIS OPTION IS
C CROSSWIND TRANSMISSION AS A FUNCTION OF DOWNWIND DISTANCE AT
C A SINGLE GIVEN TIME RATHER THAN TRANSMISSION AS A FUNCTION
C OF TIME AT A SINGLE GIVEN LINE OF SIGHT.

C NEW INPUTS ARE ENTERED THROUGH THE "NAME" CARD AS:
C NAME
C STIME = SINGLE GIVEN TIME AT WHICH SCREEN IS TO BE SAMPLED
C FRONT = LENGTH OF SCREEN TO BE SAMPLED (ALONGWIND)
C DELX = INCREMENTS BETWEEN CONTIGUOUS LINES OF SIGHT
C MCOOPT = OPTION TO SUPPRESS INTERMEDIATE OUTPUT (1 = SUPPRESS)

C ALSO THE TIME OF DETONATION FOR EACH MUNITION IS REQUIRED (IN
C SECONDS) AS THE FOURTH ENTRY ON THE "MUNC" CARD

C ***** THE CAVEATS FOR THE NEW OPTION ARE:
C 1) THE LINES OF SIGHT MUST BE CROSSWIND
C 2) STARTING POINT FOR SAMPLING IS THE OBSERVER-TARGET
C COORDINATES ENTERED ON THE "OBSC" AND "TARC" CARDS
C 3) THE PRNT OPTION ON THE "OUTP" CARD MUST BE ZERO
C 4) A PLOT FILE OPTION HAS BEEN ADDED AS THE FOURTH ENTRY ON
C THE "OUTP" CARD (NPLT=1 WILL CREATE A PLOT FILE OF THE
C FINAL RESULTS ON UNIT NPLOTU - SEE EOMAIN)

C NOTE: THE FOLLOWING COMMON BLOCK ALLOWS MUNITION BURN DURATION
C AND OBSCURATION PERIODS UP TO 16.0 MINUTES (960 SEC)
C TO ALTER PERIOD, CHANGE MAXS AND DIMENSIONS OF SMAS, PVOL AND
C CLTOT.
COMMON /MO5/ SMAS(960),PVOL(960),CLTOT(960),SMTRAN(7),RI(9),
*EXTC(8),ZL(2),XL(2),YL(2),XINT(2),YINT(2),ZINT(2),IFLAG(2)
DATA MAXS/960/
DATA QDIV /0.,.3,.50,.60,.65,.70,.80,1.0,1.,1.,1.,1./
SMK00090
SMK00100
SMK00110
SMK00120
SMK00130
SMK00140
SMK00150
SMK00160
SMK00170
SMK00180
SMK00190

C NOTE: FOR SHORTER OR LONGER BURNS AND OBSCURATIONS, USERS MAY
C SUBSTITUTE FOR THE COMMON BLOCK AND MAXS PARAMETER AND ALTER
C DIMENSIONS FOR SMAS, PVOL AND CLTOT.
SMK00200
SMK00210
SMK00220
SMK00230
SMK00240
SMK00250

C SMOKE MODEL
C FORMAL INPUTS:
C WAVE1 = WAVELENGTH IN MICROMETERS (OR 94. GHZ) FROM ALLOWED
C BANDS (SEE EXTC RECORD BELOW).
C ICLMAT = CLIMATOLOGY FLAG. IF 1 THEN /CLYMAT/ VALUES OVER-
C RIDE METR RECORD VALUES.
C IGEOSW = GEOMETRY FLAG FROM COMMON GEOMET. IF SET
C TO 1 IN EOMAIN, THEN OBSERVER AND TARGET
C COORDINATES ARE PASSED TO SMOKE, AND ANGLE
C XNORTH SET TO 90. DEGREES.
C FORMAL OUTPUTS:
C TRANS = TRANSMISSION AT TIME ETO (DEFAULT) OR USER
C SPECIFIED TIME (SEE OUTP RECORD BELOW) FOR
C WAVELENGTH WAVE1 ALONG THE OBSERVER-TARGET
C LINE-OF-SIGHT.
C IERR = ERROR FLAG. 1 IF ERROR IN SMOKE.
C USER RECORDS INPUT:
C EACH CARD BEGINS WITH A 4 LETTER IDENTIFIER IN COL 1-4,
C FOLLOWED BY AS MANY (REAL) FIELDS AS NEEDED, 10 COL.
SMK00260
SMK00270
SMK00280
SMK00290
SMK00300
SMK00310
SMK00320
SMK00330
SMK00340
SMK00350
SMK00360
SMK00370
SMK00380
SMK00390
SMK00400
SMK00410
SMK00420

```

PER FIELD BEGINNING IN COL 11. THE CARDS ARE NOT ORDER
DEPENDENT ALTHOUGH VALUES ON EACH RECORD MUST FOLLOW THE ORDER
SHOWN IN COMMENTS BELOW. FORMAT (2A2,6X,7F10.3)

THE FOLLOWING ARE REQUIRED RECORDS FOR AT LEAST ONE INPUT SET.

IDENT.	VARS.	DESCRIPTION	
MUNC	XM,YM,ZM,TM (M,M,M,SEC)	MUNITION COORDINATES AND EVENT TIME	SMK00430 SMK00440 SMK00450 SMK00460 SMK00470 SMK00480 SMK00490 SMK00500 SMK00510 SMK00520
BART	STO ETO DIO ANGLX	COMPUTATION TIMES AND X-AXIS DEFINITION: OUTPUT STARTING TIME (SEC. SINCE IGNITION) ENDING TIME FOR CALCULATION (SEC.) TIME INCREMENT FOR OUTPUT TABLES (SEC) ANGLE OF POSITIVE X-AXIS WRT NORTH (DEG. CLOCKWISE WRT NORTH) ASSUMED 90 DEG. IF IGEOSW IS 1 FROM EOMAIN	SMK00530 SMK00540 SMK00550 SMK00560 SMK00570 SMK00580 SMK00590 SMK00600 SMK00610 SMK00620 SMK00630 SMK00640 SMK00650 SMK00660 SMK00670 SMK00680 SMK00690

OPTIONAL DEPENDING ON PARAMETERS CHOSEN:

IDENT.	VARS.	DESCRIPTION	
OBSC	XO,YO,ZO	OBSERVER COORDINATES (IGNORED IF IGEOSW IS 1 FROM EOMAIN). (M,M,M)	SMK00660 SMK00670 SMK00680 SMK00690
TARC	XT,YT,ZT	TARGET COORDINATES (IGNORED IF IGEOSW IS 1 FROM EOMAIN). (M,M,M)	SMK00700 SMK00710 SMK00720 SMK00730 SMK00740 SMK00750 SMK00760 SMK00770 SMK00780 SMK00790 SMK00800 SMK00810 SMK00820 SMK00830 SMK00840 SMK00850 SMK00860 SMK00870 SMK00880 SMK00890 SMK00900 SMK00910 SMK00920
OUTP		OPTIONAL RECORD TO SELECT AMOUNT OF PRINT OUTPUT AND TO SELECT CRITERIA FOR RETURNED TRANSMISSION.	
PRNT		IF 0., ALL FULL OUTPUT LISTINGS CREATED. IF 1. IS ENTERED AT ANY POINT, THEN ALL FURTHER OUTPUT IS SUPPRESSED, EXCEPT FOR THE FINAL ACCUMULATED EFFECTS LISTING OF TOTAL CL AND TOTAL TRANSMITTANCE FOR COMBINED MULTIPLE INPUT SETS. (DEFAULT IS PRNT = 0.)	
CRITER		SELECTS CHOICE OF TRANSMISSION RETURNED FROM SMOKE. 0. = RETURN TOTAL TRANSMITTANCE COMPUTED AT LAST TIME ETO. (DEFAULT CASE) 1. = RETURN THE MINIMUM VALUE OF TOTAL TRANSMITTANCE COMPUTED FOR WAVELENGTH WAVE1. 2. = RETURN VALUE OF TOTAL TRANSMITTANCE FOR WAVELENGTH WAVE1 COMPUTED AT USER- SPECIFIED TIME TIMTRN BELOW.	
TIMTRN		REQUIRED ONLY IF CRITER IS 2., TIME FOR WHICH TRANSMITTANCE RETURNED IS COMPUTED. SHOULD BE CLOSE OR EQUAL TO A TABLE TIME (AS DETERMINED BY THE BART RECORD) FOR ACCURACY.	
IPLT		PLOT CODE ADDED IN ORDER TO PLOT OUTPUT ON UNIT NPLTU IF IPLT = 1.	
MUNT		REQUIRED IF BURN RECORD IS NOT USED. OTHERWISE, OPTIONAL. ANY NON-ZERO VALUES INPUT WILL OVER- RIDE PREVIOUS SOURCE DEFINITIONS (INCLUDING THOSE FROM THE BURN RECORD.)	SMK00930 SMK00940 SMK00950 SMK00960 SMK00970 SMK00980 SMK00990 SMK01000 SMK01010 SMK01020 SMK01030 SMK01040 SMK01050 SMK01060 SMK01070 SMK01080 SMK01090
XN		NUMBER OF MUNITIONS IGNITED AT THE SAME LOCATION AND AT THE SAME TIME (DIMENSIONLESS)	
FW		FILL WEIGHT OF ONE MUNITION (LBS.) FOR WP,PWP,HC OR RP (BUT RATE OF BURN IN GAL./HR. FOR FOG OIL. NOTE 1 GAL/HR= 0.93 G/S)	
TBURN		BURN DURATION FOR THIS MUNITION (SEC)	
TYPE		TYPE OF SMOKE (DIMENSIONLESS) 1.=WP, 2.=PWP OR WP WICK/WEDGE, 3.=HC, 4.=FOG OIL, 5.=RP	
EFF		MUNITION BURN EFFICIENCY. (PERCENT)	

YF IF INPUT AS 0., THEN DEFAULT IS USED. SMK01100
 YIELD FACTOR (DIMENSIONLESS). IF 0., DEFAULTS SMK01110
 TO RELATIVE HUMIDITY DEPENDENT STRAIGHT SMK01120
 LINE FIT FOR WP, PWP, RP FROM JOHNSON SMK01130
 AND FORNEY. SIMILARLY FOR HC. FOG OIL IS SMK01140
 SET TO 1. SMK01150
 METR REQUIRED IF ICLMAT IS ZERO. OTHERWISE, SMK01160
 NEEDED ONLY FOR E OR F PASQUILL CATEGORY TO SMK01170
 PROVIDE TGRAD WHICH IS NOT AVAILABLE IN SMK01180
 /CLYMAT/. SMK01190
 RELHUM RELATIVE HUMIDITY (PERCENT) SMK01200
 UW WIND VELOCITY (M/SEC) SMK01210
 WDIR WIND DIRECTION (USUAL MET CONVENTION, SMK01220
 ANGLE IN DEG. CLOCKWISE FROM NORTH OF SMK01230
 DIRECTION FROM WHICH WIND ORIGINATES.) SMK01240
 PCAT PASQUILL CATEGORY (DIMENSIONLESS) SMK01250
 1.-A, 2.-B, 3.-C, 4.-D, 5.-E, 6.-F SMK01260
 AIRT SURFACE AIR TEMPERATURE (DEG C) SMK01270
 TGRAD VERT TEMP GRADIENT (C DEG/M). SMK01280
 EXAMPLE: TGRAD=(AIRT(10 M)-AIRT(.5 M))/9.5 M SMK01290
 (USED ONLY FOR PASQUILL CATEGORIES E, F) SMK01300
 EXTC OPTIONAL USER OVERRIDE FOR EXTINCTION SMK01310
 COEFFICIENTS. SMK01320
 IF RECORD NOT USED, OR FOR ANY VALUES READ SMK01330
 IN AS 0., THE EXTINCTION COEFF. SMK01340
 DEFAULTS TO ALPHA ARRAY VALUE IN STRANS. SMK01350
 INPUT EXTINCTION COEFF. (M**2/G) ORDER ON CARD SMK01360
 CORRESPONDS TO THE BANDS: SMK01370
 0.4-0.7 MICROMETERS SMK01380
 0.7-1.2 MICROMETERS SMK01390
 1.06 MICROMETERS SMK01400
 3.0-5.0 MICROMETERS SMK01410
 8.0-12. MICROMETERS SMK01420
 10.6 MICROMETERS SMK01430
 94.0 GHZ. SMK01440
 BURN OPTIONAL - SELECTS BUILT-IN MUNITION CHARACTER- SMK01450
 ISTICS FROM THE BRATE ROUTINE FOR ONE (XN=1) SMK01460
 MUNITION. VALUES ARE FOR FILL WEIGHT (FW), BURN SMK01470
 DURATION (TBURN), SMOKE TYPE (ITYPE), EFFICIENCY SMK01480
 (EFF). YIELD FACTOR IS SET TO ZERO SO THAT RH SMK01490
 MODEL DEPENDENT VALUES ARE USED. ANY VALUES READ SMK01500
 IN AS NON-ZERO ON A MUNT RECORD (WHICH IS SMK01510
 OPTIONAL IF A BURN CARD IS USED) WILL OVERRIDE SMK01520
 THE DEFAULTS STORED IN BRATE. SMK01530
 TYPM MUNITION TYPE: SMK01540
 0. = USER DEFINED MUNITION SOURCE CHAR. SMK01550
 1. = 155MM HC, M1 CANISTER. SMK01560
 2. = 155MM HC, M2 CANISTER. SMK01570
 3. = 105MM HC CANISTER. SMK01580
 4. = 155MM HC M116B1 PROJ. SMK01590
 5. = 105MM HC M84A1 PROJ. SMK01600
 6. = SMOKE POT HC M5 SMK01610
 7. = 60MM WP M302 CARTRIDGE SMK01620
 8. = 81MM WP M375A2 SMK01630
 9. = 4.2 IN WP M328A1 SMK01640
 10. = 155MM WP M110E2 SMK01650
 11. = 105MM WP M60A2 SMK01660
 12. = 4.2 IN PWP M328A1 SMK01670
 13. = 5. IN PWP ZUNI MK4 SMK01680
 14. = 2.75 IN WP WEDGE SUB-MUNITION. SMK01690
 15. = 3. IN WP WICK SUB-MUNITION SMK01700
 16. = 6. IN WP WICK SUB-MUNITION SMK01710
 17. = 155MM WP WEDGE XM825 (92 SUB-MUN.) SMK01720
 18. = 81MM RP WEDGE NAVY SUB-MUNITION SMK01730


```

* 2HPW,2HP,2HOR,2H W,2HP,2HWI,2HCK,2H/W,2HED,2HGE,2H,
* 2HHC,2H S,2HMO,2HKE,2H,2H,2H,2H,2H,2H,2H,
* 2HFO,2HG,2HOI,2HL,2HSM,2HOK,2HE,2H,2H,2H,2H,
* 2HRE,2HD,2HPH,2HQS,2HPH,2HOR,2HUS,2H,2HRP,2H,2H,
DATA IR,2HME,2HTR,2HMU,2HNT,2HBA,2HRT,2HMU,2HNC,2HOB,2HSC,2HTA,
*2HRC,2HEX,2HTC,2HGO,2H,2HBU,2HRN,2HMI,2HSC,2HOU,2HTP,2HDO,
*2HNE,2HNA,2HME /
DATA BRAT1,BRAT2,BRAT3,BRAT4,BRAT5/1.,0.,0.,0.,1./
DATA TYPM,XN,FW,TBURN,EFF,YF,RELHUM,UW,WDIR,ANGLX,PCAT,
*AIR,TGRAD/13*0./
DATA ITGRAD/0/
BFUNCT)=T*(BRAT1+T*(BRAT2/2.+T*(BRAT3/3.+T*BRAT4/4.)))
**BRAT5*.07818288*ALOG(1.+358800.*T)
NUMDIV=8
NRUNS=0
TIMTRN=1.
ICRTR=0
TRANS=1.
MCUOPT=0
NOPRNT=0
NOMORE=0
NCY=0
DO 1 J=1,MAXS
CLTOT(J)=0.
CONTINUE
DO 2 J=1,8
5 2 EXTCT(J)=0.
CONTINUE
NRUNS=NRUNS+1
MUNRD=0
KWAWE=0
IF (KWAWE1.GE.0.4.AND.WAVE1.LT.0.7) KWAWE=1
IF (KWAWE1.GE.0.7.AND.WAVE1.LE.1.2) KWAWE=2
IF (KWAWE1.GE.3.0.AND.WAVE1.LE.5.0) KWAWE=4
IF (KWAWE1.GE.8.0.AND.WAVE1.LE.12.0) KWAWE=5
IF (KWAWE1.GT.1.059.AND.WAVE1.LT.1.061) KWAWE=3
IF (KWAWE1.GT.10.59.AND.WAVE1.LT.10.61) KWAWE=6
IF (KWAWE1.GT.93.9.AND.WAVE1.LT.94.1) KWAWE=7
IF (KWAWE1.GT.3188.AND.WAVE1.LT.3195.) KWAWE=7
IF (KWAWE.EQ.0) GOTO 998
C*** BEGINNING OF READ LOOP
NCHK=0
DO 70 I = 1, 15
IF (I.EQ.15) GO TO 310
READ(10IN,20) IR1,IR2,(R1(J),J=1,7)
20 FORMAT(2A2,6X,7F10,3)
C*** RELATING INPUT DATA TO VARIABLE NAMES.
IF (IR1.EQ.IR(1).AND.IR2.EQ.IR(2)) GOTO 90
IF (IR1.EQ.IR(3).AND.IR2.EQ.IR(4)) GOTO 100
IF (IR1.EQ.IR(5).AND.IR2.EQ.IR(6)) GOTO 110
IF (IR1.EQ.IR(7).AND.IR2.EQ.IR(8)) GOTO 120
IF (IR1.EQ.IR(9).AND.IR2.EQ.IR(10)) GOTO 130
IF (IR1.EQ.IR(11).AND.IR2.EQ.IR(12)) GOTO 140
IF (IR1.EQ.IR(13).AND.IR2.EQ.IR(14)) GOTO 150
IF (IR1.EQ.IR(15).AND.IR2.EQ.IR(16)) GOTO 155
IF (IR1.EQ.IR(17).AND.IR2.EQ.IR(18)) GOTO 105
IF (IR1.EQ.IR(19).AND.IR2.EQ.IR(20)) GOTO 70
IF (IR1.EQ.IR(21).AND.IR2.EQ.IR(22)) GOTO 115
IF (IR1.EQ.IR(23).AND.IR2.EQ.IR(24)) GOTO 154
IF (IR1.EQ.IR(25).AND.IR2.EQ.IR(26)) GOTO 121
WRITE(10OUT,80)
80 FORMAT(1H,72HINVALID DATA CARD-DOES NOT CONFORM TO PROPER CO
*NVENTION IN SMOKE ROUTINE)
WRITE(10OUT,30) IR1,IR2,(R1(J),J=1,7)
30 FORMAT(1H,2A2,6X,7F10,3)
GO TO 999
90 IF (ICLMAT.EQ.1) GOTO 92
IF (R1(1).NE.0.) RELHUM= R1(1)
IF (R1(2).NE.0.) UW = R1(2)
IF (R1(3).NE.0..OR.R1(2).NE.0.) WDIR = R1(3)

```

	IF (R1(4).NE.0.) PCAT = R1(4)	SMK03130
	ICAT=IFIX(PCAT+.0001)	SMK03140
92	AIRT = R1(5)	SMK03150
	TGRAD = R1(6)	SMK03160
	ITGRAD=1	SMK03170
	NCHK=1	SMK03180
	GO TO 70	SMK03190
100	IF (R1(1).NE.0.) XN = R1(1)	SMK03200
	IF (R1(2).NE.0.) FW = R1(2)	SMK03210
	IF (R1(3).NE.0.) TBURN = R1(3)	SMK03220
	IF (R1(4).NE.0.) ITYPE = IFIX(R1(4)+.0001)	SMK03230
	IF (MUNRD.EQ.0.OR.R1(5).NE.0.) EFF = R1(5)	SMK03240
	YF=R1(6)	SMK03250
	MUNRD=1	SMK03260
	NCHK=1	SMK03270
	GO TO 70	SMK03280
105	TYPM=R1(1)	SMK03290
	CALL BRATE(IERR,MUNRD,TYPM,XN,FW,TBURN,ITYPE,EFF,YF,BRAT1,	SMK03300
	*BRAT2,BRAT3,BRAT4,BRAT5)	SMK03310
	IF (IERR.NE.0) WRITE (IOOUT,95) TYPM	SMK03320
95	FORMAT(37H IN SMOKE, ILLEGAL MUNITION TYPE READ ,F5.0)	SMK03330
	IF (IERR.NE.0) GOTO 999	SMK03340
	IF (R1(2).EQ.0..AND.R1(3).EQ.0..AND.R1(4).EQ.0..AND.R1(5).EQ.0.)	SMK03350
	GOTO 93	SMK03360
	*BRAT1=R1(2)	SMK03370
	BRAT2=R1(3)	SMK03380
	BRAT3=R1(4)	SMK03390
	BRAT4=R1(5)	SMK03400
	BRAT5=R1(6)	SMK03410
93	IF (TYPM.GT.0.) MUNRD=1	SMK03420
	NCHK=1	SMK03430
	GO TO 70	SMK03440
110	ISTO = IFIX(R1(1)+.0001)	SMK03450
	IETO = IFIX(R1(2)+.0001)	SMK03460
	IDTO = IFIX(R1(3)+.0001)	SMK03470
	ANGLX=R1(4)	SMK03480
	NCHK=1	SMK03490
	GO TO 70	SMK03500
115	NOPRNT=0	SMK03510
	IF (R1(1).NE.0.) NOPRNT=1	SMK03520
	CRITER=R1(2)	SMK03530
	ICRTR=IFIX(CRITER+.001)	SMK03540
	IF (ICRTR.GT.2) ICRTR=2	SMK03550
	IF (ICRTR.LT.0) ICRTR=0	SMK03560
	IF (ICRTR.EQ.2) TIMTRN=R1(3)	SMK03570
	IPLT=IFIX(R1(4))	
	GOTO 70	SMK03580
120	XM = R1(1)	SMK03590
	YM = R1(2)	SMK03600
	ZM = R1(3)	SMK03610
	TM = R1(4)	
	NCHK=1	SMK03620
	GO TO 70	SMK03630
121	MODE=1	
	STIME = R1(1)	
	ISRN = IFIX(STIME+0.0001)	
	FRONT = R1(2)	
	DELX = R1(3)	
	IF (DELX.LE.0.0) DELX=5.0	
	XXX=FRONT/DELX	
	NPTS=IFIX(XXX)+1	
	MCUOPT=IFIX(R1(4))	
	GO TO 70	
130	XO = R1(1)	SMK03640
	YO = R1(2)	SMK03650
	ZO = R1(3)	SMK03660
	NCHK=1	SMK03670
	GO TO 70	SMK03680
140	XT = R1(1)	SMK03690
	YT = R1(2)	SMK03700

```

      ZT      = R1(3)
      NCHK=1
      GO TO 70
150  DO 152 J=1,7
152  EXTC(J)=R1(J+1)
      EXTC(8)=0.
70  CONTINUE
154  NOMORE=1
      IF (NCHK.EQ.0) NRUNS=NRUNS-1
      IF (NCHK.EQ.0) GOTO 980
155  CONTINUE
C*****REDEFINE MUNITION EFFICIENCY IF INPUT AS ZERO
      IF (EFF.GT.0.0) GO TO 11
      IF (ITYPE.EQ.1) EFF=100.
      IF (ITYPE.EQ.2) EFF=65.0
      IF (ITYPE.EQ.3) EFF=40.0
      IF (ITYPE.EQ.4) EFF=100.0
      IF (ITYPE.EQ.5) EFF=50.
11  CONTINUE
      IF (ICLMAT.NE.1) GOTO 12
      RELHUM=RH
      UW      =WINDVEL
      WNDIR=WNDIR
      ICAT    =IPASCT
      PCAT=FLOAT(ICAT)
      AIRT    =TEMP
12  CONTINUE
      IF (IGEOSW.NE.1) GOTO 13
      DISKTM=1000.
C...  CONVERT UNITS FROM KM TO M.
      XT=PTS(1)*DISKTM
      YT=PTS(2)*DISKTM
      ZT=PTS(3)*DISKTM
      XO=PTS(4)*DISKTM
      YO=PTS(5)*DISKTM
      ZO=PTS(6)*DISKTM
      ANGLX=90.
13  CONTINUE
      IF (ITYPE.LT.1.OR.ITYPE.GT.5) IERR=1
      IF (IERR.EQ.1) WRITE (IOOUT,180) ITYPE
180  FORMAT(1X,31HIN SMOKE, INVALID SMOKE TYPE = ,I4)
      IF (IERR.EQ.1) GOTO 999
C***  CHECK BURN RATE FOR 100 PERCENT BURN AT TBURN...
      IF (BRAT2.EQ.0..AND.BRAT3.EQ.0..AND.BRAT4.EQ.0..AND.BRAT5.EQ.0.)
      *BRAT1=1.
      VNORM=BFUNC(1.)
      BRAT1=BRAT1/VNORM
      BRAT2=BRAT2/VNORM
      BRAT3=BRAT3/VNORM
      BRAT4=BRAT4/VNORM
      BRAT5=BRAT5/VNORM
      IF (XN.LE.0.) XN=1.
      IF (ITYPE.EQ.1.AND.TBURN.GT.1.) TBURN=1.
C***  SET UP EXTINCTION COEFF TO BE USED...
      CALL STRANS(CL,SMTRAN,ITYPE,EXTC,0)
      CALL SMASPP(XN,EFF,FW,RELHUM,W,ITYPE,YF)
      TGR=TGRAD
      TGRAD=ABS(TGRAD)
      IF (NOPRNT.EQ.1.AND.NRUNS.GT.1) GOTO 255
      IF (MCUOPT.EQ.1.AND.NRUNS.GT.1) GOTO 255
      IF (NRUNS.GT.1) WRITE (IOOUT,172) NRUNS
      IF (NRUNS.EQ.1) WRITE (IOOUT,1720) NRUNS
172  FORMAT(1H1,50X,17(1H*)/51X,1H*,15X,1H*/51X,1H*,5X,5HSMOKE,5X,
      *1H*,15X,9HEXECUTION ,13/51X,1H*,15X,1H*/51X,17(1H*)/)
1720  FORMAT(1H0,50X,17(1H*)/51X,1H*,15X,1H*/51X,1H*,5X,5HSMOKE,5X,
      *1H*,15X,9HEXECUTION ,13/51X,1H*,15X,1H*/51X,17(1H*)/)
C***  REPORTING INPUT DATA.
      IF (ICAT.GE.5.AND.ITYPE.NE.4.AND.ITGRAD.EQ.0) WRITE (IOOUT,98) TGRAD
98  FORMAT(1X,44HIN SMOKE ROUTINE PASQUILL CATEGORIES E AND F,
      *29H REQUIRE TEMPERATURE GRADIENT/10X,23HIF SMOKE IS EXOTHERMIC.,

```

```

*20H A VALUE OF TGRAD = ,F7.2,25H C DEG/M WILL BE ASSUMED./) SMK04380
WRITE(IOOUT,170) SMK04390
170 FORMAT(5X,15HSMOKE MUNITIONS,22X,25HMETEOROLOGICAL CONDITIONS,25X, SMK04400
123HEXTINCTION COEFFICIENTS) SMK04410
WRITE(IOOUT,190) (ITTL(J,ITYPE),J=1,11),UW,EXTC(1) SMK04420
190 FORMAT(3X,11A2,11X,10HWINDSPEED,14X,F5.1,2X,3HM/S,16X, SMK04430
120H0.4-0.7 MICROMETERS ,F7.3,2X,7HM**2/GM) SMK04440
WRITE(IOOUT,200) XN,EXTC(2) SMK04450
200 FORMAT(3X,10HNO. ROUNDS,1X,F5.0,17X,22HWIND DIRECTION (USUAL ,28X, SMK04460
120H0.7-1.2 MICROMETERS ,F7.3,2X,7HM**2/GM) SMK04470
IF (ITYPE.NE.4) WRITE(IOOUT,210) FW,WDIR,EXTC(3) SMK04480
210 FORMAT(3X,11HFILL WEIGHT,F8.3,3H LB,11X,23HMET CONVENTION AZIMUTH)SMK04490
1,F6.1,2X,7HDEGREES,12X,20H1.06 MICROMETERS ,F7.3,2X,7HM**2/GM) SMK04500
IF (ITYPE.EQ.4) WRITE(IOOUT,215) FW,WDIR,EXTC(3) SMK04510
215 FORMAT(3X,9HBURN RATE,2X,F6.1,2X,6HGA/LHR, 8X,23HMET CONVENTION AZSMK04520
11MUTH> ,F6.1,2X,7HDEGREES,12X,20H1.06 MICROMETERS ,F7.3,2X, SMK04530
27HM**2/GM) SMK04540
WRITE(IOOUT,220)TBURN,RELHUM,EXTC(4) SMK04550
220 FORMAT(3X,9HBURN TIME,F8.1,2X,3HSEC,11X,17HRELATIVE HUMIDITY,7X, SMK04560
1F5.1,2X,7HPERCENT,12X,20H3.0-5.0 MICROMETERS ,F7.3,2X,7HM**2/GM) SMK04570
LCAT=JPASCT(ICAT) SMK04580
WRITE(IOOUT,230) EFF,LCAT,EXTC(5) SMK04590
230 FORMAT(3X,10HEFFICIENCY,1X,F6.1,2X,7HPERCENT,7X,17HPASQUILL CATEGOSMK04600
1RY,9X,A1,23X,20H8.0-12. MICROMETERS ,F7.3,2X,7HM**2/GM) SMK04610
WRITE (IOOUT,235) YF,AIRT,EXTC(6) SMK04620
235 FORMAT(3X,12HYIELD FACTOR,F6.2,15X,15HAIR TEMPERATURE,8X,F6.1,2X, SMK04630
18HDEGREE C,11X,20H10.6 MICROMETERS ,F7.3,2X,7HM**2/GM) SMK04640
WRITE(IOOUT,240)TGRAD,EXTC(7) SMK04650
240 FORMAT(36X,14HTEMP. GRADIENT,10X,F6.2,9H C DEG./M,11X, SMK04660
111H94.0 GHZ,9X,F7.3,2X,7HM**2/GM) SMK04670
IF (ICAT.GE.5.AND.TGR.LT.0.) WRITE(IOOUT,250) SMK04680
250 FORMAT(36X,24H(ASSUMED POSITIVE INPUT)) SMK04690
255 NRAT=0 SMK04700
IF (BRAT1.NE.0.) NRAT=1 SMK04710
IF (BRAT2.NE.0.) NRAT=2 SMK04720
IF (BRAT3.NE.0.) NRAT=3 SMK04730
IF (BRAT4.NE.0.) NRAT=4 SMK04740
IF (NOPRNT.EQ.1.AND.NRUNS.GT.1) GOTO 258 SMK04750
IF (MCUOPT.EQ.1.AND.NRUNS.GT.1) GOTO 258
IF (NRAT.EQ.1) WRITE (IOOUT,175) BRAT1 SMK04760
IF (NRAT.EQ.2) WRITE (IOOUT,176) BRAT1,BRAT2 SMK04770
IF (NRAT.EQ.3) WRITE (IOOUT,177) BRAT1,BRAT2,BRAT3 SMK04780
IF (NRAT.EQ.4) WRITE (IOOUT,178) BRAT1,BRAT2,BRAT3,BRAT4 SMK04790
IF (NRAT.EQ.0) WRITE (IOOUT,181) SMK04800
181 FORMAT(40X,19HBURN RATE PROFILE =) SMK04810
IF (BRAT5.NE.0.) NRAT=5 SMK04820
NPWP=0 SMK04830
IF ((ITYPE.EQ.2.OR.ITYPE.EQ.5).AND.NRAT.GT.1) NPWP=1 SMK04840
IF (NRAT.EQ.5) WRITE (IOOUT,179) BRAT5 SMK04850
179 FORMAT(40X,2H+ ,F8.4,41H*0.0781829*(358800./((1.+358800.*T/TBURN)))SMK04860
175 FORMAT(1H0,40X,19HBURN RATE PROFILE =,F8.4) SMK04870
176 FORMAT(1H0,31X,19HBURN RATE PROFILE =,F8.4,2H +, SMK04880
*F8.4,10H (T/TBURN)) SMK04890
177 FORMAT(1H0,20X,19HBURN RATE PROFILE =,F8.4,2H +, SMK04900
*F8.4,12H (T/TBURN) +,F8.4,13H (T/TBURN)**2) SMK04910
178 FORMAT(1H0, 9X,19HBURN RATE PROFILE =,F8.4,2H+ , SMK04920
*F8.4,12H (T/TBURN) +,F8.4,15H (T/TBURN)**2 +,F8.4, SMK04930
*13H (T/TBURN)**3) SMK04940
C*** PROVIDE COORDINATES... SMK04950
258 ANGL=ANGLX+180.-WDIR SMK04960
IF (ANGL.GT.360.) ANGL=ANGL-360. SMK04970
IF (ANGL.LT.0.) ANGL=ANGL+360. SMK04980
IF (NOPRNT.EQ.1.AND.NRUNS.GT.1) GOTO 285 SMK04990
XP1=XQ-XM SMK05000
XP2=XT-XM SMK05010
YP1=YQ-YM SMK05020
YP2=YT-YM SMK05030
ZPP1=ZQ-ZM SMK05040
ZPP2=ZT-ZM SMK05050
CA=COS(ANGL*PIRAD) SMK05060

```

```

SA=SIN(ANGL*PIRAD)
XPP3=0.
YPP3=0.
ZPP3=0.
XPP1=XP1*CA+YP1*SA
YPP1=YP1*CA-XP1*SA
XPP2=XP2*CA+YP2*SA
YPP2=YP2*CA-XP2*SA
IF (MCUOPT.EQ.1.AND.NRUNS.GT.1) GO TO 285
WRITE (IOOUT,260)
WRITE (IOOUT,270) XM,YM,ZM,XPP3,YPP3,ZPP3,X0,Y0,Z0,XPP1,YPP1,
12PPI,XT,YT,ZT,XPP2,YPP2,ZPP2
260 FORMAT(1H0,30X,17HFIELD COORDINATES,20X,43HROTATED COORD.(WIND X-AS
1XIS,MUNITION ORIGIN)/27X,3H(X),8X,3H(Y),8X,3H(Z),20X,4H(XW),
27X,4H(YW),7X,4H(ZW))
270 FORMAT(1X,22HMUNITION COORDINATES=,3(F9.2,2X),6HMETERS,6X,
13(F9.2,2X),6HMETERS/1X,22HOBSEVER COORDINATES=,3(F9.2,2X),
26HMETERS,6X,3(F9.2,2X),6HMETERS/1X,22HTARGET COORDINATES=,
33(F9.2,2X),6HMETERS,6X,3(F9.2,2X),6HMETERS)
WRITE (IOOUT,275) ANGLX, TM
275 FORMAT(1X,48HANGLE OF ORIGINAL X-AXIS, CLOCKWISE WRT NORTH = ,
*F7.2,5H DEG.,/,5X,12HEVENT TIME =,F6.1,4H SEC)
IF(MODE.GT.0) WRITE(IOOUT,1000) MODE,STIME,FRONT,DELX
1000 FORMAT(5X,5HMODE=,12,/,5X,12HSCREEN TIME=,F6.1,/,5X,6HFRONT=,F6.1,
*,5X,11HINCREMENT=,F6.1)
IF (NOPRNT.EQ.1.OR.MCUOPT.EQ.1) GO TO 285
IF(MODE.EQ.0) WRITE(IOOUT,280)
IF(MODE.GT.0) WRITE(IOOUT,281)
280 FORMAT(1H0,6X,4HTIME,3X,6HLENGTH,3X,5HWIDTH,4X,6HHEIGHT,2X,
110HPATHLENGTH,4X,2HCL,24X,12HTRANSMISSION,/,7X,5H(SEC),3(1X,8H(METERS
2ERS)),2X,8H(METERS),2X,9H(GM/M**2),12X,28HSPECTRAL BANDS (MICROMETERS
3ERS),/,60X,39H0.4-0.7 0.7-1.2 1.06 3.0-5.0 8.0-12.,
415H 10.6 94.GHZ,/)
281 FORMAT(1H0,6X,5H*LOS*,3X,6HLENGTH,3X,5HWIDTH,4X,6HHEIGHT,2X,10HPAT
1HLENGTH,4X,2HCL,24X,12HTRANSMISSION,/,4X,8H(METERS),3(1X,8H(METERS
2)),2X,8H(METERS),2X,9H(GM/M**2),12X,28HSPECTRAL BANDS (MICROMETERS
3),/,60X,40H0.4-0.7 0.7-1.2 1.06 3.0-5.0 8.0-12.,
415H 10.6 94.GHZ,/)
C*** BEGINNING OF CALCULATIONS.
285 IF (W.EQ.0.) GOTO 999
IF (ICAT.LT.1.OR.ICAT.GT.6) GOTO 999
IF(IDTO.EQ.0) IDTO = 1
IF (ISTO.LE.0) ISTO=IDTO
IF (IETO.LE.ISTO) IETO=ISTO
C*** CALCULATIONS DEPENDENT ON TIME.
C*** SET COMPUTATION TIME STEP DTIME TO 1 SEC. FOR HC/FOG OIL/PWP/
C RP AND WP WICKS/WEDGES
C*** BUT TO TABLE REPORT TIME INCREMENT FOR WP
DTIME=FLOAT(IDTO)
TRANS=1.0
TSUB(1)=0.
TSUB(2)=TBURN
IF (NPWP.EQ.0) GOTO 620
IS=IFIX(5.*TBURN+.0001)
IF (IS.LT.1) GOTO 610
DO 286 I=2,NUMDIV
TSUB(I)=1./5.
286 CONTINUE
DO 600 I=1,IS
FI=FLOAT(I)/5.
T=FI/TBURN
TMS=BFUN(T)
DO 601 JDIV=2,NUMDIV
IF (TMS.LE.0DIV(JDIV)) TSUB(JDIV)=FI
601 CONTINUE
TSUB(NUMDIV)=TBURN
600 CONTINUE
GOTO 620
610 NPWP=0
620 CONTINUE

```



```

PRNTOP=NOPRNT
IF (MODE.GT.0) PRNTOP=MCUOPT
IF (PRNTOP.NE.1) WRITE(IOOUT,290) XWRIT,X,YFULL,Z,PATHL,CL,
290 *(SMTRAN(K),K=1,7)
FORMAT(6X,F5.0,2X,F6.0,3X,F6.0,3X,F6.0,4X,F6.2,4X,F7.2,7(3X,F5.3))
IF (ICRTR.EQ.2.AND.TV.GT.(TIMTRN+.5)) GOTO 7
IF (ICRTR.EQ.1.AND.SMTRAN(KWAVE).GE.TRANS) GOTO 7
TRTM=TV
TRANS=SMTRAN(KWAVE)
7 CONTINUE
IF (NCY.GE.MAXS) GOTO 6
NCY=NCY+1
CLTOT(NCY)=CLTOT(NCY)+CL
6 CONTINUE
700 CONTINUE
C*** FINAL OUTPUT
IF (NOMORE.EQ.0) GOTO 5
IF (NCY.LE.0) GOTO 999
IF (NRUNS.LE.1.AND.NPWP.EQ.0) GOTO 997
IF (MODE.EQ.0) WRITE(IOOUT,982) NRUNS
IF (MODE.GT.0) WRITE(IOOUT,984) NRUNS
982 FORMAT(1H1,40X,19HCOMBINED EFFECT OF ,13,21H EXECUTIONS IN SMOKE:/
*10X,10HTIME (SEC),5X,11HCL (G/M**2),21X,12HTRANSMISSION/
*10X,10(1H-),5X,11(1H-),4X,
*54H0.4-0.7 0.7-1.2 1.06 3.0-5.0 8.0-12. 10.6 94.GHZ/)
984 FORMAT(1H1,40X,19HCOMBINED EFFECT OF ,13,21H EXECUTIONS IN SMOKE:/
*10X,11HLOS(METERS),4X,11HCL (G/M**2),21X,12HTRANSMISSION/
*10X,10(1H-),5X,11(1H-),4X,
*54H0.4-0.7 0.7-1.2 1.06 3.0-5.0 8.0-12. 10.6 94.GHZ/)
TRANS=1.0
DO 985 I=1,NCY
CL=CLTOT(I)
TU=FLOAT(ISTO+(I-1)*IDTO)
CALL STRANS(CL,SMTRAN,ITYPE,EXTC,1)
YWRIT=TU
IF (MODE.GT.0) YWRIT=XPP0+XM+(I-1)*DELX
WRITE (IOOUT,983) YWRIT,CL,(SMTRAN(J),J=1,7)
IF (IPLT.EQ.1) WRITE(NPLOU,883) YWRIT,CL
883 FORMAT(1X,F6.1,2X,F8.3)
983 FORMAT(12X,F6.0,8X,F8.2,6X,7(1X,F5.3,2X))
IF (ICRTR.EQ.2.AND.TO.GT.(TIMTRN+.5)) GOTO 985
IF (ICRTR.EQ.1.AND.SMTRAN(KWAVE).GE.TRANS) GOTO 985
TRTM=TO
TRANS=SMTRAN(KWAVE)
985 CONTINUE
997 CONTINUE
WRITE (IOOUT,3100) WAVE1,TRANS,TRTM
TRANS=SMTRAN(KWAVE)
RETURN
998 WRITE (IOOUT,3200)
GOTO 999
3100 FORMAT(1H0,5X,37H***TRANSMISSION RETURNED TO MAIN FOR
1 14HWAVELENGTH OF ,F8.3,16H MICROMETERS IS ,F5.3,
2 8H AT TIME ,F7.0)
3200 FORMAT(1H0,10X,35HINVALID WAVELENGTH PASSED FROM MAIN
1 //,10X,27H TRANS=1.0 RETURNED TO MAIN,/)
310 CONTINUE
C*** ERROR CONDITION
WRITE(IOOUT,320)
320 FORMAT(1H ,105HMORE THAN 13 DATA CARDS HAVE BEEN INPUT . PLEASE CH
*ECK THAT THERE ARE NO MORE THAN 13 DATA CARDS PER RUN.)
999 CONTINUE
IERR=1
TRANS=1.0
RETURN
END

```

SMK06070
SMK06080
SMK06090
SMK06100
SMK06110
SMK06120
SMK06130
SMK06140
SMK06150
SMK06160
SMK06170
SMK06180
SMK06190
SMK06200
SMK06210
SMK06220
SMK06230
SMK06240
SMK06250
SMK06260
SMK06270
SMK06280
SMK06290
SMK06300
SMK06310
SMK06320
SMK06330
SMK06340
SMK06350
SMK06360
SMK06370
SMK06380
SMK06390
SMK06400
SMK06410
SMK06420
SMK06430
SMK06440
SMK06450
SMK06460
SMK06470
SMK06480
SMK06490
SMK06500
SMK06510
SMK06520
SMK06530
SMK06540
SMK06550
SMK06560
SMK06570
SMK06580
SMK06590
SMK06600
SMK06610

```

SUBROUTINE CLSMOK( ICODE, CLGAUS, ITYPE, CL, W, PATHL, TBURN, TBRN, PCNT, CLS00010
* TSTAGE, NPWP, XLIM, YLIM, ZLIM, TO, TLIM, DTIME, X, Y, Z, BRAT1, BRAT2, CLS00020
* BRAT3, BRAT4, BRAT5) CLS00030
COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK CLS00040
COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU CLS00050
NOTE: THE FOLLOWING COMMON BLOCK ALLOWS MUNITION BURN DURATION CLS00060
AND OBSCURATION PERIODS UP TO 16.0 MINUTES (960 SEC) CLS00070
COMMON /MOS/ SMAS(960),PVOL(960),CLTOT(960),SMTRAN(7),R1(9), CLS00080
*EXTC(8),ZL(2),XL(2),YL(2),XINT(2),YINT(2),ZINT(2),IFLAG(2) CLS00090
NOTE: TO CHANGE MAXIMUM BURN OR OBSCURATION DURATION, CHANGE CLS00100
PARAMETER MAXS AND DIMENSIONS OF SMAS,PVOL,CLTOT TO MAXS CLS00110
THIS SUBROUTINE CALCULATES THE VOLUME OF THE SMOKE CLOUD AND CLS00120
THE CONCENTRATION * LENGTH(CL) AT THE INTERSECTION OF THE OBSERVER CLS00130
TARGET LOS OF THE CLOUD FORMED BY THE TOTAL NUMBER OF SMOKE CLS00140
MUNITIONS WHICH WERE DETONATED AT THE SAME TIME AND FROM THE SAME CLS00150
POINT. FOR CONTINUOUS TYPE BURNS (HC, FOG OIL, PWP, RP, AND CLS00160
WP WICKS OR WEDGES), THE BURN MASS INCREMENT AND PATH LENGTH CLS00170
TO VOLUME RATIO ARE STORED FOR EACH PUFF. THESE PUFFS ARE CLS00180
SUBSEQUENTLY ADDED TO FIND THE TOTAL EFFECT OF OBSCURANT. CLS00190
INPUTS CLS00200
ICODE A FLAG TO BE SET BY USER TO 0 ON FIRST CALL CLS00210
TO PROGRAM, WHICH WILL THEN RESET IT TO 1. CLS00220
ITYPE SMOKE TYPE 1=WP, 2=PWP OR WP WICK/WEDGE CLS00230
3=HC, 4=FOG OIL, 5=RP CLS00240
W SMOKE MASS PRODUCED BY XN MUNITIONS (G FOR CLS00250
TYPES 1-3 AND 5, G/S FOR TYPE 4) CLS00260
PATHL PATHLENGTH OF SMOKE CLOUD AS IT INTERSECTS THE CLS00270
OBSERVER-TARGET LINE OF SIGHT (M) CLS00280
TBURN TOTAL LENGTH OF TIME OF BURN (S) CLS00290
NPWP FLAG FOR PWP/RP/WP WICKS OR WEDGES. IF NON-ZERO, CLS00300
THEN BURN IS IN 25 PERCENT (OF W) STAGES. CLS00310
TBRN PARTIAL BURN DURATION THIS STAGE (SEC) CLS00320
TSTAGE START OF PARTIAL BURN (SEC) CLS00330
XLIM,YLIM,ZLIM DIMENSIONS (LENGTH, HALF-WIDTH, HEIGHT) CLS00340
OF CLOUD AT END OF EXOTHERMIC RISE TIME (M) CLS00350
TO TIME AFTER IGNITION (S) CLS00360
TLIM TIME OF TERMINATION OF HEAT RISE (S) CLS00370
DTIME TIME INCREMENT OF COMPUTATION (S) CLS00380
ENDTIM ENDING TIME OF COMPUTATION (S) CLS00390
X,Y,Z CURRENT LEADING EDGE CO-ORDINATES (M) CLS00400
BRAT1,2,3,4 POLYNOMIAL BURN RATE COEFF. CLS00410
CLGAUS UNIT CONTRIBUTION FROM GAUSSIAN PUFF. CLS00420
OUTPUTS CLS00430
CL COMPUTED CL IN (G/M**2) FOR THIS MUNITION SET. CLS00440
C*** SIMPLE FOR INSTANTANEOUS BURN OF WP CLS00450
BFUN(T)=T*(BRAT1+T*(BRAT2/2.+T*(BRAT3/3.+T*(BRAT4/4.)))+BRAT5* CLS00460
*0.07818288*ALOG(1.+358800.*T) CLS00470
MAXS=960 CLS00480
IF (ITYPE.GT.1) GOTO 100 CLS00490
VOL=0.25*(4.*PI/3.) *X*Y*Z CLS00500
CL=W*(.25*PATHL/VOL +.75 *CLGAUS) CLS00510
RETURN CLS00520
C*** INITIALIZE CLS00530
100 CL=0. CLS00540
IF (ICODE.NE.0) GOTO 110 CLS00550
ICODE=1 CLS00560
IB=0 CLS00570
IPL=0 CLS00580
START=0. CLS00590
BRNOUT=FLOAT(IFIX(.9999+TBRN)) CLS00600
NRAT=2 CLS00610
IF (BRAT2.EQ.0..AND.BRAT3.EQ.0..AND.BRAT4.EQ.0.) NRAT=1 CLS00620
IF (BRAT5.NE.0.) NRAT=5 CLS00630
CLS00640
CLS00650
CLS00660
CLS00670
CLS00680
CLS00690
CLS00700

```

```

KSUB=0
TOTPV=0.
BR=W
IF (ITYPE.NE.4) BR=W/BRNOUT
IF (NPWP.NE.0) BR=BR*PCNT
IF (BRNOUT.LE.1.) NRAT=1
IF (NRAT.GT.1) BTHEN=BFUN(TSTAGE/TBURN)
C*** STORE PUFF MASS EMITTED AT TIME TO
110 IF (TO.GT.BRNOUT) GOTO 120
IF (IB.GE.MAXS) GOTO 120
IB=IB+1
IF (NRAT.EQ.1) GOTO 120
IF (IB.GE.MAXS) WRITE (100UT,900) IB
900 FORMAT(1X,61H*** WARNING - IN SMOKE, MAXIMUM STORAGE FOR BURN DURAC
*TION OF ,15.55H SEC IS FULL. ACCURACY BEYOND THIS POINT DECREASES.
* *** )
TM=TO
IF (TO.GT.TBRN) TM=TBRN
T=(TM+TSTAGE)/TBURN
BNOW=BFUN(T)
SMAS(IB)=W*(BNOW-BTHEN)
BTHEN=BNOW
IF (ITYPE.EQ.4) SMAS(IB)=SMAS(IB)*TBURN
C*** COMPUTE VOLUME AT TIME TO OF FIRST PUFF
120 IF (ITYPE.EQ.3) GOTO 130
IF (ITYPE.EQ.4) GOTO 140
C*** PWP, RP OR WP WICKS/WEDGES CLOUD
VOL=0.25*(4.*PI/3.)*X*Y*Z
GOTO 200
130 IF (TO.LE.TLIM) GOTO 140
C*** POST RISE REGION HC CONE.
VOL=0.5*(PI/3.)*XLIM*YLIM*ZLIM
C*** POST-RISE FRUSTRUM OF APPROXIMATED ELLIPTIC CONE.
XZPROJ=Z*(X-XLIM)/(Z-ZLIM)
VFRUST=0.5*(PI/3.)*(XZPROJ*(Y*Z-YLIM*ZLIM)+(X-XLIM)*YLIM*ZLIM)
VOL=VOL+VFRUST
GOTO 200
C*** HC BEFORE THE END OF EXOTHERMIC RISE AND FOG OIL CASE
140 VOL=0.5*(PI/3.)*X*Y*Z
C*** STORE PATH LENGTH/VOLUME RATIO AT TIME TO FOR FIRST EXPANDING
C UNIFORM AND GAUSSIAN PUFF CONTRIBUTION.
200 IF (ITYPE.NE.4) PV=.25*(PATHL/VOL)+.75*CLGAUS
IF (ITYPE.EQ.4) PV=PATHL/VOL
IF (PV.LE.0.) AND (START.EQ.0.) RETURN
IF (PV.GT.0.) AND (START.EQ.0.) START=TO
IF (IPL.GE.MAXS) GOTO 300
IPL=IPL+1
IF (IPL.EQ.MAXS) WRITE (100UT,910) TO,IPL
910 FORMAT(1X,58H*** WARNING - IN SMOKE, MAX. STORAGE FOR CLOUD VOLUME
*S OF ,15.34H SEC. IS FULL. COMPUTATION TIME = ,F6.0,5H SEC./
*15X,40HACCURACY BEYOND THIS POINT DECREASES ***)
PVOL(IPL)=PV
C*** SUM CL FOR PUFFS
300 IF (NRAT.EQ.1) GOTO 400
LMIN=MAX(1,(IPL-IB+1))
IF (IPL.LT.LMIN) RETURN
N=0
DO 320 J=LMIN,IPL
K=IPL-J+1
CDEL=SMAS(K)*PVOL(J)
CL=CL+CDEL
N=N+1
IF (N.LT.120) GOTO 320
IF (ABS(CDEL).LT.1.E-5) RETURN
320 CONTINUE
RETURN
C*** FAST COMPUTATION FOR CONSTANT BURN RATES
400 TOTPV=TOTPV+PVOL(IPL)
C INDEX OF LAST PUFF EMITTED
IF (IPL.LT.MAXS) KSUB=IPL-IB

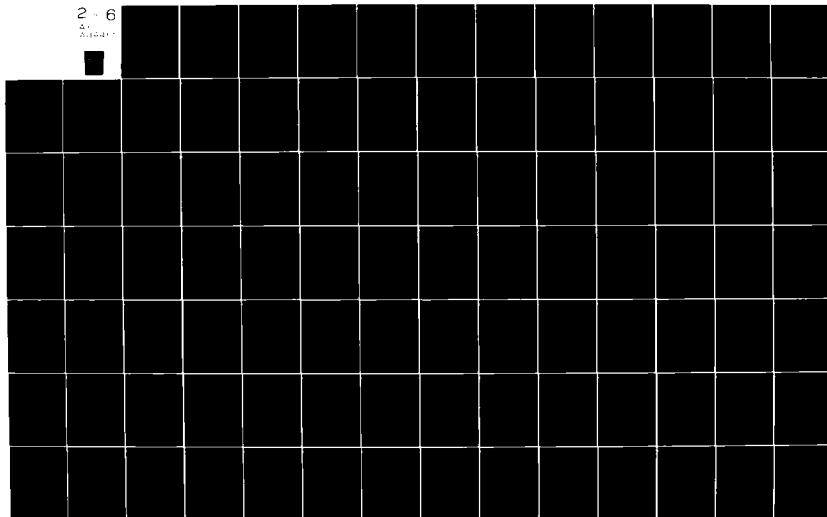
```

C KEEP REMOVING LAST PUFF IF POSSIBLE	CLS01410
IF (IPL.EQ.MAXS) KSUB=KSUB+1	CLS01420
IF (KSUB.GT.IPL) KSUB=IPL	CLS01430
C ONLY REMOVE PUFFS AFTER BURN HAS STOPPED (IE LAST PUFF OUT)	CLS01440
IF (KSUB.GT.0) TOTPV=TOTPV-PVOL(KSUB)	CLS01450
CL=TOTPV*BR	CLS01460
RETURN	CLS01470
END	CLS01480

AD-A114 417 ARMY ELECTRONICS RESEARCH AND DEVELOPMENT COMMAND WS--ETC F/G 4/1
PROGRAM LISTINGS FOR EOSAEL 80-B AND ANCILLARY CODES AGAUS AND --ETC(11)
FEB 82 R G STEINHOFF
UNCLASSIFIED ERADCOM/ASL-TR-0107-V2-SU NL

2 - 6

AD-A114 417



```

120 TB=TBRN
IF (ITYPE.EQ.1.AND.TBRN.GT.1.) TB=1.
IF (NPWP.NE.0) TB=TB/PCNT
C*** IF NON-EXOTHERMIC, RETURN.
IF (ITYPE.EQ.4) RETURN
EN=800.0
IF (ITYPE.EQ.3) EM=500.0
IF (ITYPE.EQ.5) EM=660.
F=0.00001*3.59*453.59*EM*FW*XM*(EFF/100.)/TB
C*** CALCULATION OF CLOUD HEIGHT (EXO) COEFFICIENT FOR UNSTABLE ATM.
C*** CONDITIONS (A,B,C)
C3=1.6/UW*(F**0.3333)
IF (ITYPE.EQ.3) GO TO 200
IF (ICAT.EQ.4) GO TO 131
IF (ICAT.GT.4) GO TO 141
TLIM=0.0
C*** DETERMINE TIME AND TOTAL CLOUD HEIGHT AT TERMINATION OF HEAT RISE.
123 TLIM=TLIM+2.
X=UW*TLIM
Y=(9.1+C1*X)/2.
V=.75*(2.*3.14159)**1.5*(Y/2.15)**2*(2.73+C3*X)/4.3
C=W/(V*TB)
IF (C.GT.0.11) GO TO 123
HLIM=C3*X**0.667
RETURN
C*** CALCULATION OF CLOUD HEIGHT (EXO) COEFFICIENTS FOR NEUTRAL ATM.
C*** CONDITIONS (D).
131 XS=10.0*F**0.4
CNEUT=1.6*(F**0.3333)*XS**0.667/UW
RETURN
141 CONTINUE
C*** CALCULATION OF CLOUD HEIGHT (EXO) COEFFICIENT AND TOTAL CLOUD
C*** HEIGHT AT TERMINATION OF HEAT RISE FOR STABLE ATM. CONDITIONS (E,F)
SBAR=9.8/(AIRT+273.0)*(TGRAD+0.0098)
HLIM=1.4*(F/(UW*SBAR))**0.333
RETURN
C *** CALCULATE PARAMETERS FOR HC 10 SEC. RISE TIME.
200 TLIM=10.
XLIM=TLIM*UW
YLIM=(9.1+C1*XLIM)/2.
ZLIM=2.73+C2*XLIM
IF (ICAT.EQ.4) GOTO 231
HLIM=C3*XLIM**0.667
ZLIM=ZLIM+HLIM
IF (ICAT.LT.5) RETURN
SBAR=9.8/(AIRT+273.0)*(TGRAD+0.0098)
HTEST=1.4*(F/(UW*SBAR))**0.333
IF (HLIM.LE.HTEST) RETURN
C*** IF UNSTABLE ATMOSPHERE REACHES MAX. BEFORE 10 SEC. COMPUTE
C*** TLIM.
XLIM=(HTEST/C3)**1.5
TLIM=XLIM/UW
YLIM=(9.1+C1*XLIM)/2.
HLIM=HTEST
ZLIM=2.73+C2*XLIM+HLIM
RETURN
231 XS=10.0*F**0.4
CNEUT=C3*(XS**0.667)
HLIM=CNEUT*(0.4+0.64*(XLIM/XS)+2.2*(XLIM/XS)**2)/
*(1.+0.8*XLIM/XS)**2
ZLIM=ZLIM+HLIM
RETURN
END

```

```

SC000710
SC000720
SC000730
SC000740
SC000750
SC000760
SC000770
SC000780
SC000790
SC000800
SC000810
SC000820
SC000830
SC000840
SC000850
SC000860
SC000870
SC000880
SC000890
SC000900
SC000910
SC000920
SC000930
SC000940
SC000950
SC000960
SC000970
SC000980
SC000990
SC001000
SC001010
SC001020
SC001030
SC001040
SC001050
SC001060
SC001070
SC001080
SC001090
SC001100
SC001110
SC001120
SC001130
SC001140
SC001150
SC001160
SC001170
SC001180
SC001190
SC001200
SC001210
SC001220
SC001230
SC001240
SC001250
SC001260
SC001270
SC001280
SC001290
SC001300
SC001310
SC001320
SC001330

```

```

SUBROUTINE SMASSP(XN,EFF,FW,RELHUM,W,ITYPE,YF)
COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUS
SMOKE MODEL
THIS SUBROUTINE CALCULATES THE SMOKE MASS PRODUCED BY XN
MUNITIONS AT THE SAME POINT AND SAME TIME.
INPUTS  ITYPE SMOKE TYPE
        1 WHITE PHOSPHOROUS
        2 PLASTICIZED WHITE PHOSPHORUS OR WP WICK WEDGE
        3 HC
        4 FOG OIL
        5 RED PHOSPHORUS
XN      NUMBER OF MUNITIONS AT THIS POINT AND TIME.
RELHUM  RELATIVE HUMIDITY (PERCENT)
EFF     MUNITION EFFICIENCY (PERCENT)
FW      FILL WEIGHT (LBS FOR TYPES 1-3, GAL/HR FOR TYPE 4)
W       SMOKE MASS PRODUCED FOR XN MUNITIONS.
        G FOR TYPES 1-3 AND 5, G/S FOR TYPE 4
YF      OPTIONAL USER SUPPLIED YIELD FACTOR

W=0.0
IF(ITYPE.LT.1.OR.ITYPE.GT.5) GO TO 90
GO TO (101,101,102,103,101),ITYPE
90  WRITE(IOOUT,50)ITYPE
50  FORMAT(//,26H ERROR IN SMASSP:SMOKE TYPE ,I4,10H UNDEFINED)
    RETURN
101 Y=.028*RELHUM+3.4
    CONVER=453.592
    GO TO 110
102 Y=1.17+.014*RELHUM
    CONVER=453.592
    GO TO 110
C***  FOR FOG OIL, W IS A RATE, WHERE FW IS IN GAL/HR AND .93
C***  CONVERTS TO G/S.
103 Y=1.0
    CONVER=0.93
110 IF(YF.NE.0.) Y=YF
    W=XN*Y*FW*CONVER*(EFF/100.0)
    RETURN
END

```



```

SUBROUTINE STRANS(CL, SMTRAN, ITYPE, EXTC, ICALL)
SMOKE MODEL      CLOUD TRANSMISSION
THIS SUBROUTINE CALCULATES TRANSMISSION IN 7 SPECTRAL REGIONS
(0.4-0.7, 0.7-1.2, 1.06, 3.0-5.0, 8.0-12.0, 10.6 MICROMETERS AND 94.GHZ)
FOR A GIVEN SMOKE TYPE AND CONCENTRATION LENGTH
INPUTS  ITYPE  SMOKE TYPE
        1  WHITE PHOSPHOROUS
        2  PLASTICIZED WHITE PHOSPHORUS, WP WICK WEDGE
        3  HC
        4  FOG OIL
        5  RED PHOSPHORUS
CL      COMPUTED CL IN (G/M**2)
ICALL   = 0 SETS UP EXTC ARRAY USED FOR COMPUTATIONS AND
        = 1 EXECUTES TRANSMISSION CALCULATION.
OUTPUTS SMTRAN TRANSMISSION THROUGH SMOKE (DECIMAL)
        EXTC  ARRAY OF EXTINCTION COEFF. ACTUALLY USED
        IN TRANSMISSION CALCULATION. EXTC(8) IS
        USED AS A FLAG FOR ICALL=0 REPLACEMENT.

        IF EXTC(8)=ITYPE, NO CHANGES ARE MADE IN EXTC ARRAY IE.,
        ALPHA VALUES DO NOT REPLACE EXTC VALUES.
        IF EXTC(8)=0, ONLY THOSE VALUES IN EXTC WHICH ARE
        ZERO ARE REPLACED BY THE STORED VALUES IN
        ALPHA (ITYPE COLUMN).
        IF EXTC(8) IS NOT 0, OR ITYPE, THEN ALL EXTC VALUES
        ARE REPLACED BY CORRESPONDING ALPHA VALUES AND
        EXTC(8) IS SET TO ITYPE.

        DIMENSION ALPHA(7,5), SMTRAN(7), EXTC(8)
        DATA ALPHA /4.304,2.166,1.541,0.350,0.338,0.364,0.001,
        *           4.304,2.166,1.541,0.350,0.338,0.364,0.001,
        *           4.579,2.186,2.040,0.190,0.052,0.051,0.001,
        *           6.851,4.592,3.497,0.245,0.020,0.018,0.001,
        *           4.304,2.166,1.541,0.350,0.338,0.364,0.001/
C*** TRANSMISSION CALCULATED BY BEER'S LAW APPROXIMATION
C*** IF ICALL=0, EXTC ARRAY IS FORMED OR MODIFIED...
        IF (ICALL.NE.0) GOTO 20
        IF (EXTC(8).EQ.FLOAT(ITYPE)) GOTO 18
        IF (EXTC(8).EQ.0.) GOTO 15
        DO 13 J=1,7
        13 EXTC(J)=0.
        15 EXTC(8)=FLOAT(ITYPE)
        DO 17 J=1,7
        17 IF (EXTC(J).EQ.0.) EXTC(J)=ALPHA(J,ITYPE)
        18 CONTINUE
        19 RETURN
C*** FOR ICALL NON-ZERO, COMPUTE TRANSMISSION USING EXTC VALUES.
        DO 30 I=1,7
        20 SMTRAN(I)=EXP(-EXTC(I)*CL)
        30 CONTINUE
        RETURN
END

```

```

SUBROUTINE WGGEOM(ICALL,CLGAUS,ITYPE,XPP1,YPP1,ZPP1,XPP2,YPP2,
1ZPP2,C1,C2,C3,TO,UW,ICAT,HLIM,TLIM,CNEUT,XS,PATHL,X,Y,Z,XLIM,
2YLIM,ZLIM)
NOTE: THE FOLLOWING COMMON BLOCK ALLOWS MUNITION BURN DURATION
AND OBSCURATION PERIODS UP TO 16.0 MINUTES (960 SEC)
COMMON /MO5/ SMAS(960),PVOL(960),CLTOT(960),SMTRAN(7),R1(9),
*EXTC(8),ZL(2),XL(2),YL(2),XINT(2),YINT(2),ZINT(2),IFLAG(2)
COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK
DATA MAXS /960/
NOTE: TO CHANGE LENGTH OF BURN OR OBSCURATION, RESET
MAXS AND REDIMENSION SMAS, PVOL AND CLTOT ARRAYS
SMOKE MODEL GEOMETRY
INPUT:
ITYPE = SMOKE TYPE CODE
XPP1,YPP1,ZPP1 = OBSERVER IN MUNITION CENTERED COORDINATES
YPP1,YPP2,ZPP2 = TARGET IN MUNITION CENTERED COORD. (X AXIS ALONG
WIND VECTOR.)
C1,C2,C3 = CLOUD GROWTH PARAMETERS (SEE PROGRAM SCONST)
TO = TIME SINCE DETONATION (SECONDS)
UW = WIND SPEED (METERS/SECOND)
ICAT=PASQUILL CATEGORY
HLIM,TLIM,CNEUT,XS= EXOTHERMIC RISE PARAMETERS
ICALL = FLAG SET BY USER TO 0 FOR 1ST CALL. RESET BY PROGRAM
TO 1 THEREAFTER.
OUTPUT:
X = CLOUD LENGTH (METERS)
Y = CLOUD HALF-WIDTH (METERS)
Z = CLOUD HEIGHT (METERS)
PATHL = PATHLENGTH OF LOS THROUGH SMOKE CLOUD
SUBROUTINES CALLED... DIRECTLY: XYZINT, GPUFF, INDIRECTLY: QROOT
*** TRANSLATION TO PLACE MUNITION AT (0,0,0)
IF (ICALL.NE.0) GOTO 10
ICALL=1
KCALL=0
C*** CALCULATE LEADING EDGE LOCATION AT TO.
10 A=UW*TO
B=(9.1/2.0)+A*C1/2.
IF (ITYPE.EQ.3.OR.ITYPE.EQ.4) GOTO 33
C*** WP/PWP/RP COMPUTATION.
GO TO (11,11,11,21,31,31),ICAT
11 CONTINUE
C=2.73+C2*A+C3*A**0.667
CLIM=2.73+C2*A+HLIM
IF (TO.GT.TLIM.AND.C.GT.CLIM)C=CLIM
GO TO 41
21 CONTINUE
C=2.73+C2*A
IF (A.LE.XS) C=C+C3*A**0.667
IF (A.GT.XS) C=C+CNEUT*(0.4+0.64*(A/XS)+2.2*(A/XS)**2)/
(1.0+0.8*(A/XS))**2
GO TO 41
31 CONTINUE
C=2.73+C2*A+C3*A**0.667
CLIM=2.73+C2*A+HLIM
IF (C.GT.CLIM)C=CLIM
GO TO 41
C*** HC, FOG OIL COMPUTATION.
33 C=2.73+C2*A
IF (ITYPE.EQ.4) GOTO 41
IF (TO.LT.TLIM) GOTO 35
C=C+HLIM
GO TO 41
35 IF (ICAT.NE.4.OR.A.LE.XS)C=C+C3*A**0.667
IF (ICAT.EQ.4 .AND. A.GT.XS)

```

WGG00010
WGG00020
WGG00030
WGG00040
WGG00050
WGG00060
WGG00070
WGG00080
WGG00090
WGG00100
WGG00110
WGG00120
WGG00130
WGG00140
WGG00150
WGG00160
WGG00170
WGG00180
WGG00190
WGG00200
WGG00210
WGG00220
WGG00230
WGG00240
WGG00250
WGG00260
WGG00270
WGG00280
WGG00290
WGG00300
WGG00310
WGG00320
WGG00330
WGG00340
WGG00350
WGG00360
WGG00370
WGG00380
WGG00390
WGG00400
WGG00410
WGG00420
WGG00430
WGG00440
WGG00450
WGG00460
WGG00470
WGG00480
WGG00490
WGG00500
WGG00510
WGG00520
WGG00530
WGG00540
WGG00550
WGG00560
WGG00570
WGG00580
WGG00590
WGG00600
WGG00610
WGG00620
WGG00630
WGG00640
WGG00650
WGG00660
WGG00670
WGG00680
WGG00690
WGG00700

```

1 C=C+CNEUT*(0.4+0.64*(A/XS)+2.2*(A/XS)**2)/
1<1.0+0.8*(A/XS)**2
41 CONTINUE
C*** NOW COMPUTE PATH LENGTHS -- SELECT ELLIPSOID FOR PHOSPHORUS
C*** OR ELLIPTICAL CONES FOR HC, FOG OIL.
C*** FIRST GAUSSIAN PUFF AND IMAGE PUFF CONTRIBUTION, THEN FULLY MIXED
NCODE=2
IF (ITYPE.EQ.3.OR.ITYPE.EQ.4) NCODE=1
IF (ITYPE.NE.4)
*CALL GPUFF(KCALL,CLGAUS,A,B,C,C2,XPP1,XPP2,YPP1,YPP2,ZPP1,ZPP2)
X=A
Y=B
Z=C
IF (ITYPE.EQ.3) GOTO 200
100 X0=A
Y0=B
Z0=C
150 XL(1)=XPP1
XL(2)=XPP2
YL(1)=YPP1
YL(2)=YPP2
ZL(1)=ZPP1
ZL(2)=ZPP2
CALL XYZINT(NCODE,XL,YL,ZL,X0,Y0,Z0,XINT,YINT,ZINT,IFLAG)
PATHL=SQRT((XINT(1)-XINT(2))**2+(YINT(1)-YINT(2))**2+
*(ZINT(1)-ZINT(2))**2)
IF (ITYPE.EQ.3) GOTO 220
RETURN
200 ICONE=1
IF (TO.LE.TLIM) GOTO 100
ICONE=0
C*** FOR HC, BREAK PATH UP INTO TWO PARTS. FIRST RISE PORTION CONE,
C THEN POST-RISE FRUSTRUM.
X0=XLIM
Y0=YLIM
Z0=ZLIM
GOTO 150
220 IF (ICONE.EQ.1) RETURN
ICONE=1
C*** NOW POST RISE PORTION,
IF (PATHL.EQ.0.) GOTO 100
IF (IFLAG(1).NE.3.AND.IFLAG(2).NE.3) RETURN
IF (IFLAG(1).EQ.3.AND.IFLAG(2).EQ.3) RETURN
K=1
IF (IFLAG(2).EQ.3) K=2
J=1
IF (XL(2).LT.XL(1)) J=2
XL(J)=XINT(K)
YL(J)=YINT(K)
ZL(J)=ZINT(K)
X0=A
Y0=B
Z0=C
CALL XYZINT(NCODE,XL,YL,ZL,X0,Y0,Z0,XINT,YINT,ZINT,IFLAG)
PATHL=PATHL+SQRT((XINT(1)-XINT(2))**2+(YINT(1)-YINT(2))**2+
*(ZINT(1)-ZINT(2))**2)
RETURN
END

```

```

WGG00710
WGG00720
WGG00730
WGG00740
WGG00750
WGG00760
WGG00770
WGG00780
WGG00790
WGG00800
WGG00810
WGG00820
WGG00830
WGG00840
WGG00850
WGG00860
WGG00870
WGG00880
WGG00890
WGG00900
WGG00910
WGG00920
WGG00930
WGG00940
WGG00950
WGG00960
WGG00970
WGG00980
WGG00990
WGG01000
WGG01010
WGG01020
WGG01030
WGG01040
WGG01050
WGG01060
WGG01070
WGG01080
WGG01090
WGG01100
WGG01110
WGG01120
WGG01130
WGG01140
WGG01150
WGG01160
WGG01170
WGG01180
WGG01190
WGG01200
WGG01210
WGG01220
WGG01230
WGG01240
WGG01250
WGG01260
WGG01270
WGG01280

```

```

      FUNCTION QROOT(ISIGN,A,B,C)
C*****PURPOSE:
C      TO FIND ROOTS OF A QUADRATIC EQUATION.
      IF (A.EQ.0.) GOTO 2
      XX=1.0*ISIGN
      TEST= B*B - 4.0*A*C
      IF (TEST.LT.0.0) GOTO 1
      QROOT=(-1.0*B + XX*SQRT(TEST))/(2.0*A)
      GO TO 100
1     QROOT=0.0
      GOTO 100
2     IF (B.EQ.0.) GOTO 1
      QROOT=-C/B
100  RETURN
      END

```

```

QRO00010
QRO00020
QRO00030
QRO00040
QRO00050
QRO00060
QRO00070
QRO00080
QRO00090
QRO00100
QRO00110
QRO00120
QRO00130
QRO00140
QRO00150

```

```

SUBROUTINE XYZINT(NCODE,XL,YL,ZL,X0,Y0,Z0,XINT,YINT,ZINT,IFLAG)
DIMENSION XL(2),YL(2),ZL(2),XINT(2),YINT(2),ZINT(2),IFLAG(2)
DIMENSION DIST(2),TEST(2)
C*****SUBROUTINE XYZINT*****
C*****PURPOSE:
TO FIND THE X,Y,Z INTERCEPTS OF A TARGET-OBSERVER LINE OF SIGHT
WITH A SMOKE CLOUD DESCRIBED BY:
NCODE = 1 A HALF ELLIPTIC CONE WITH APEX AT THE ORIGIN
AND LEADING EDGE TRUNCATION BY THE X=X0 PLANE
AND BOTTOM EDGE TRUNCATION BY THE Z=0 PLANE.
NCODE = 2 A QUARTER ELLIPSOID WITH APEX
AT THE ORIGIN AND WITH LEADING EDGE TRUNCATED BY
THE X=X0 AND BOTTOM EDGE TRUNCATED BY THE Z=0 PLANE.
C*****INPUT:
XL(2),YL(2),ZL(2) = X,Y,Z COORDINATES OF TWO POINTS THROUGH
WHICH THE LOS PASSES (IE. TARGET AND OBSERVER COORDINATES).
X0,Y0,Z0 = LENGTH OF SEMI-AXES OF ELLIPSOID.
C*****OUTPUT:
XINT(2),YINT(2),ZINT(2) = X,Y,Z COORDINATES OF THE INTERCEPTS
OF THE LOS WITH THE ELLIPSOID.
IFLAG(2) - INTERCEPT TYPE FOR EACH INTERCEPT COORD:
= 0 NO INTERCEPT
= 1 INTERIOR TO VOLUME
= 2 ON CONICAL OR ELLIPTICAL SURFACE
= 3 ON LEADING EDGE OF SURFACE
C*****MATHEMATICAL APPROACH:
THE EQUATION OF THE ELLIPSOID CAN BE WRITTEN AS:
 $((X-X0)/X0)**2 + ((Y/Y0)**2 + ((Z/Z0)**2 = 1$ 
AND THE EQUATION OF THE LOS CAN BE WRITTEN AS:
 $(X-XL1)/(XL2-XL1) = (Y-YL1)/(YL2-YL1) = (Z-ZL1)/(ZL2-ZL1)$ 
THE TWO EQUATIONS ARE COMBINED TO FORM A QUADRATIC EQUATION
WHICH IS SOLVED TO GIVE THE INTERCEPTS.
SIMILARLY FOR THE LOS EQUATIONS AND ELLIPTIC CONE :
 $((Z/Z0)**2 + ((Y/Y0)**2 - ((X/X0)**2 = 0$ 
C*****SPECIAL NOTES
(1) WHEN TWO OR MORE COORDINATES ARE THE SAME, SPECIAL CASES ARE
FORMED WHICH MUST BE DEALT WITH SEPARATELY BECAUSE OF
SINGULARITIES IN THE LOS EQUATION.
(2) WHEN TARGET AND/OR OBSERVER ARE INSIDE THE CLOUD INTERCEPTS
ARE TAKEN AS THE TARGET AND/OR OBSERVER COORDINATES.
(3) PROPER ACCOUNT IS TAKEN FOR A LOS INTERCEPTING THE CLOUD
LEADING EDGE BUT...
(4) ALL COORDINATES MUST BE ABOVE THE Z=0 PLANE (IE. ABOVE THE
SURFACE.)
C SUBROUTINE CALLED... GROOT
C*****INITIALIZE PROGRAM VARIABLES
I1=0
I2=0
INT=0
LEAD=0
ISURF=0
K1=0
K2=0
K3=0
K4=0
TEST(1)=0.0
TEST(2)=0.0
TEST3=0.0
TEST4=0.0
DIST(1)=0.0
DIST(2)=0.0
DELX=XL(2)-XL(1)
DELY=YL(2)-YL(1)
DELZ=ZL(2)-ZL(1)
IFLAG(1)=0
IFLAG(2)=0
C*** REJECT IMMEDIATELY IF BOTH TGT/OBS BELOW Z=0.
IF (ZL(1).LT.0. AND ZL(2).LT.0.) GOTO 800
C*****DETERMINE SPECIAL CASES FOR LOS

```

```

ICASE=1
IF (ABS(DELX).GT.(.01*ABS(DELY)).AND.
* ABS(DELX).GT.ABS(.01*ABS(DELZ))) GOTO 12
ICASE=2
IF (ABS(DELY).GT.(.01*ABS(DELZ))) GOTO 12
ICASE=3
IF (DELZ.GT.1.E-2.OR.DELZ.LT.-1.E-2) GOTO 12
ICASE=4
12 CONTINUE
C*****DEFAULT SPECIAL CASE OF OBS-TAR COINCIDENT
IF(ICASE.NE.4)GO TO 14
GO TO 800
14 CONTINUE
C*****SET UP TEST TO DETERMINE IF TARGET AND/OR OBSERVER ARE IN THE
C INTERIOR OF THE CLOUD
DO 1 I=1,2
IF (NCODE.EQ.2) TEST(I)= ((XL(I)-X0)/X0)**2 + (YL(I)/Y0)**2 +
*(ZL(I)/Z0)**2-1.
IF (NCODE.EQ.1) TEST(I)=(YL(I)/Y0)**2 + (ZL(I)/Z0)**2 -
*(XL(I)/X0)**2
1 CONTINUE
IF (TEST(1).GT.0.0)GO TO 2
IF (XL(1).LT.0..OR.ZL(1).LT.0.) GOTO 2
IF (XL(1).GT.X0) GOTO 2
IF (XL(1).EQ.X0) K1=1
IF (TEST(1).EQ.0..OR.ZL(1).EQ.0.) K3=1
I1=1
2 CONTINUE
IF (TEST(2).GT.0.0)GO TO 3
IF (XL(2).LT.0..OR.ZL(2).LT.0.) GOTO 3
IF (XL(2).GT.X0) GOTO 3
IF (XL(2).EQ.X0) K2=1
IF (TEST(2).EQ.0..OR.ZL(2).EQ.0.) K4=1
I2=1
C*****IF BOTH POINTS ARE IN THE CLOUD SET INTERCEPTS EQUAL TO THE
C TARGET-OBSERVER COORDINATES AND RETURN--OTHERWISE CONTINUE
3 IF(I1.EQ.0)GO TO 4
IF(I2.EQ.0)GO TO 4
DO 5 I=1,2
IFLAG(I)=1
XINT(I)=XL(I)
YINT(I)=YL(I)
5 ZINT(I)=ZL(I)
IF (K3.EQ.1) IFLAG(1)=2
IF (K4.EQ.1) IFLAG(2)=2
IF (K1.EQ.1) IFLAG(1)=3
IF (K2.EQ.1) IFLAG(2)=3
GO TO 999
C*****IF ONLY ONE POINT IS IN CLOUD KEEP TRACK OF IT FOR LATER
4 CONTINUE
IF(I1.EQ.0)GO TO 6
INT=1
6 IF(I2.EQ.0)GO TO 7
INT=2
7 CONTINUE
IF (K1.EQ.1) LEAD=1
IF (K2.EQ.1) LEAD=2
IF (K3.EQ.1) ISURF=1
IF (K4.EQ.1) ISURF=2
C*****SET UP LOS EQUATION DEPENDING UPON CASE
GO TO (10,20,30),ICASE
C*****CASE 1
10 SX=DELX/DELX
SY=DELY/DELX
SZ=DELZ/DELX
XI=XL(1)-SX*XL(1)
YI=YL(1)-SY*XL(1)
ZI=ZL(1)-SZ*XL(1)
GO TO 101
C*****CASE 2

```

```

XYZ000710
XYZ000720
XYZ000730
XYZ000740
XYZ000750
XYZ000760
XYZ000770
XYZ000780
XYZ000790
XYZ000800
XYZ000810
XYZ000820
XYZ000830
XYZ000840
XYZ000850
XYZ000860
XYZ000870
XYZ000880
XYZ000890
XYZ000900
XYZ000910
XYZ000920
XYZ000930
XYZ000940
XYZ000950
XYZ000960
XYZ000970
XYZ000980
XYZ000990
XYZ010000
XYZ010010
XYZ010020
XYZ010030
XYZ010040
XYZ010050
XYZ010060
XYZ010070
XYZ010080
XYZ010090
XYZ010100
XYZ010110
XYZ010120
XYZ010130
XYZ010140
XYZ010150
XYZ010160
XYZ010170
XYZ010180
XYZ010190
XYZ010200
XYZ010210
XYZ010220
XYZ010230
XYZ010240
XYZ010250
XYZ010260
XYZ010270
XYZ010280
XYZ010290
XYZ010300
XYZ010310
XYZ010320
XYZ010330
XYZ010340
XYZ010350
XYZ010360
XYZ010370
XYZ010380
XYZ010390
XYZ010400

```

```

20 SX=DELX/DELY
SY=DELY/DELY
SZ=DELZ/DELY
XI=XL(1)-SX*YL(1)
YI=YL(1)-SY*YL(1)
ZI=ZL(1)-SZ*YL(1)
GO TO 101
C*****CASE 3
30 SX=DELX/DELZ
SY=DELY/DELZ
SZ=DELZ/DELZ
XI=XL(1)-SX*ZL(1)
YI=YL(1)-SY*ZL(1)
ZI=ZL(1)-SZ*ZL(1)
101 CONTINUE
C*****SET UP QUADRATIC COEFFICIENTS
IF (NCODE.EQ.1) GOTO 60
A=(SX/X0)**2 + (SY/Y0)**2 + (SZ/Z0)**2
B=2.0*((XI/X0)*(SX/X0) + (YI/Y0)*(SY/Y0) + (ZI/Z0)*(SZ/Z0)
*-(SX/X0))
C=(XI/X0)**2 + (YI/Y0)**2 + (ZI/Z0)**2 - 2.0*XI/X0
GOTO 61
60 A=(SY/Y0)**2 + (SZ/Z0)**2 -(SX/X0)**2
B=2.0*((SY/Y0)*(SY/Y0) + (SZ/Z0)*(SZ/Z0) + (SX/X0)*(XI/X0))
C=(YI/Y0)**2 + (ZI/Z0)**2 -(XI/X0)**2
61 CONTINUE
C*****DEFAULT ALL INTERCEPTS IF ROOTS ARE COMPLEX
TEST0=B*B-4.0*A*C
IF (TEST0.GE.0.0) GO TO 888
800 DO 13 I=1,2
IFLAG(I)=0
XINT(I)=0.0
YINT(I)=0.0
13 ZINT(I)=0.0
GO TO 999
C*****SOLVE QUADRATIC FOR X,Y OR Z DEPENDING ON CASE
888 GO TO (100,200,300),ICASE
100 XINT(1)=GROOT(+1,A,B,C)
XINT(2)=GROOT(-1,A,B,C)
DO 11 I=1,2
YINT(I)=YI+SY*XINT(I)
11 ZINT(I)=ZI+SZ*XINT(I)
GOTO 400
200 YINT(1)=GROOT(+1,A,B,C)
YINT(2)=GROOT(-1,A,B,C)
DO 21 I=1,2
XINT(I)=XI+SX*YINT(I)
21 ZINT(I)=ZI+SZ*YINT(I)
GO TO 400
300 ZINT(1)=GROOT(+1,A,B,C)
ZINT(2)=GROOT(-1,A,B,C)
DO 31 I=1,2
XINT(I)=XI+SX*ZINT(I)
31 YINT(I)=YI+SY*ZINT(I)
C*** TEST FOR VALID INTERCEPTS
400 I1=0
I2=0
IFLAG(1)=2
IFLAG(2)=2
IF (ZINT(1).GE.0. .AND.XINT(1).GE.0. .AND.XINT(1).LE.X0) I1=1
IF (ZINT(2).GE.0. .AND.XINT(2).GE.0. .AND.XINT(2).LE.X0) I2=1
IF (I1.EQ.0. .OR.I2.EQ.0) GOTO 450
IF (XINT(1).EQ.XINT(2). .AND.YINT(1).EQ.YINT(2). .AND.ZINT(1).EQ.
* ZINT(2)) I2=0
IF (I1.EQ.1. .AND.I2.EQ.1) GOTO 600
C*** AT LEAST ONE INTERCEPT INVALID. FIRST COMPUTE POSSIBLE Z=0. INTCP
450 GOTO (460,470,480),ICASE
460 Z=0.
IF (SZ.EQ.0.) GOTO 500
X=-ZI/SZ

```

```

XYZ01410
XYZ01420
XYZ01430
XYZ01440
XYZ01450
XYZ01460
XYZ01470
XYZ01480
XYZ01490
XYZ01500
XYZ01510
XYZ01520
XYZ01530
XYZ01540
XYZ01550
XYZ01560
XYZ01570
XYZ01580
XYZ01590
XYZ01600
XYZ01610
XYZ01620
XYZ01630
XYZ01640
XYZ01650
XYZ01660
XYZ01670
XYZ01680
XYZ01690
XYZ01700
XYZ01710
XYZ01720
XYZ01730
XYZ01740
XYZ01750
XYZ01760
XYZ01770
XYZ01780
XYZ01790
XYZ01800
XYZ01810
XYZ01820
XYZ01830
XYZ01840
XYZ01850
XYZ01860
XYZ01870
XYZ01880
XYZ01890
XYZ01900
XYZ01910
XYZ01920
XYZ01930
XYZ01940
XYZ01950
XYZ01960
XYZ01970
XYZ01980
XYZ01990
XYZ02000
XYZ02010
XYZ02020
XYZ02030
XYZ02040
XYZ02050
XYZ02060
XYZ02070
XYZ02080
XYZ02090
XYZ02100

```


	IFLAG(I)=1	XYZ02810
	IF (LEAD.EQ.INT) IFLAG(I)=3	XYZ02820
	IF (ISURF.EQ.INT) IFLAG(I)=2	XYZ02830
	GOTO 700	XYZ02840
610	CONTINUE	XYZ02850
C***	REPLACE CLOSEST INTERCEPT OUTSIDE LOS RANGE	XYZ02860
	K1=1	XYZ02870
	IF (DIST(1).LT.DIST(2)) K1=2	XYZ02880
	XINT(K1)=XL(INT)	XYZ02890
	YINT(K1)=YL(INT)	XYZ02900
	ZINT(K1)=ZL(INT)	XYZ02910
	IFLAG(K1)=1	XYZ02920
700	IF (XINT(1).GE.(X0-.001).AND.XINT(1).LE.(X0+.001)) IFLAG(1)=3	XYZ02930
	IF (XINT(2).GE.(X0-.001).AND.XINT(2).LE.(X0+.001)) IFLAG(2)=3	XYZ02940
999	RETURN	XYZ02950
	END	XYZ02960

SUBROUTINE BRATE(IERR,MUNRD,TYPM,XN,FW,TBURN,ITYPE,EFF,YF,
*BRAT1,BRAT2,BRAT3,BRAT4,BRAT5)

THIS ROUTINE PROVIDES DEFAULT MUNITION CHARACTERISTIC VALUES TO
SMOKE. SMOKE MUNITION TYPES (TYPM) ARE GIVEN IN THE COMMENTS
OF THE MAIN ROUTINE IN SMOKE

DIMENSION B1(21),B2(21),B3(21),B4(21),B5(21)

DIMENSION F(21),T(21),E(21),IT(21)

DATA B1 / .537,.631,.2218,.537,.2218,1.,1.,1.,1.,1.,1.,1.,0.,

* 0.,.521,1.631,1.808,.1204,.653,1.731,0.,1., /

DATA B2 / .476,-.4985,3.915,.476,3.915,0.,0.,0.,0.,0.,0.,0.,0.,

* 2.106,.678,-2.556,3.1012,-3.136,-2.852,3.6832,0., /

DATA B3 / 4.779,6.745,-1.7368,4.779,-1.7368,0.,0.,0.,0.,0.,

* 0.,0.,-1.11,-5.907,2.883,-2.2104,15.309,4.341,-5.3472,0., /

DATA B4 / -5.472,-6.52,-2.3995,-5.472,-2.3995,0.,0.,0.,0.,0.,

* 0.,0.,-.748,4.012,-2.008,.206,-12.872,-3.108,3.8348,0., /

DATA B5 / 11*0.,2*1.,8*0., /

DATA IT / 3,3,3,3,3,3,1,1,1,1,1,2,2,2,2,2,2,5,5,5,4 /

DATA T / 100.,70.,120.,100.,120.,900.,1.,1.,1.,1.,1.,600.,

* 600.,240.,470.,390.,721.,260.,380.,750.,900., /

DATA F / 5.46,2.69,1.65,17.19,7.50,30.,0.76,1.75,8.14,15.6,

* 3.83,8.14,13.52,.463,.139,.234,19.98,.128,.0243,

* 19.4,40.0 /

DATA E / 40.,40.,40.,40.,40.,24.,100.,100.,100.,100.,100.,

* 60.,60.,66.,71.,67.,77.,53.,55.,51.,100. /

MAXS=21

ITP=IFIX(TYPM+.0001)

IF (ITP.EQ.0) RETURN

IF (ITP.LE.MAXS) GOTO 10

IERR=1

RETURN

BRAT1=B1(ITP)

BRAT2=B2(ITP)

BRAT3=B3(ITP)

BRAT4=B4(ITP)

BRAT5=B5(ITP)

ITYPE=IT(ITP)

IF (MUNRD.NE.0) GOTO 20

XN=1.

YF=0.

TBURN=T(ITP)

EFF=E(ITP)

FW=F(ITP)

RETURN

IF (XN.EQ.0.) XN=1.

IF (TBURN.EQ.0.) TBURN=T(ITP)

IF (EFF.EQ.0.) EFF=E(ITP)

IF (FW.EQ.0.) FW=F(ITP)

RETURN

END

BRA00010
BRA00020
BRA00030
BRA00040
BRA00050
BRA00060
BRA00070
BRA00080
BRA00090
BRA00100
BRA00110
BRA00120
BRA00130
BRA00140
BRA00150
BRA00160
BRA00170
BRA00180
BRA00190
BRA00200
BRA00210
BRA00220
BRA00230
BRA00240
BRA00250
BRA00260
BRA00270
BRA00280
BRA00290
BRA00300
BRA00310
BRA00320
BRA00330
BRA00340
BRA00350
BRA00360
BRA00370
BRA00380
BRA00390
BRA00400
BRA00410
BRA00420
BRA00430
BRA00440
BRA00450
BRA00460
BRA00470
BRA00480
BRA00490
BRA00500
BRA00510
BRA00520

```

SUBROUTINE GPUFF(KCALL,CLGAUS,A,B,C,C2,XPP1,XPP2,YPP1,YPP2,ZPP1, GPU00010
*ZPP2) GPU00020
C THIS ROUTINE COMPUTES THE CL CONTRIBUTION FROM AN EXOTHERMIC, GPU00030
C BUOYANTLY RISING SMOKE CLOUD OF GAUSSIAN DISTRIBUTION AND GPU00040
C UNIT CONCENTRATION. IT IS CENTERED ON THE LEADING EDGE AT GPU00050
C COORDINATES (A,B,C-2*SIGZ) WHERE GPU00060
C INPUTS A = CLOUD DOWNWIND DISTANCE GPU00070
C B = CLOUD BASE HALF-WIDTH (AT LEADING EDGE) GPU00080
C C = CLOUD HEIGHT (AT LEADING EDGE) GPU00090
C C2 = MOMENTUM RISE COEFFICIENT. (USUALLY BRIGGS OR LIMITED GPU00100
C RISE BRIGGS) GPU00110
C XPP1,...ZPP2 COORDINATES OF TARGET AND OBSERVER IN MUNITION GPU00120
C CENTERED COORDINATE SYSTEM WITH WIND VECTOR X-AXIS GPU00130
C OUTPUT CLGAUS = CL VALUE FOR UNIT MASS (METER**-2) GPU00140
C KCALL = SET TO 1 AFTER EVERY CHANGE IN TGT/OBS COORD. GPU00150
C DIMENSION AV(2),AP(2),BP(2),CP(2) GPU00160
C CLGAUS=0. GPU00170
C IF (KCALL.NE.0) GOTO 100 GPU00180
C KCALL=1 GPU00190
C ICASE=0 GPU00200
C*** COMPUTE LOS GENERALIZED COORDINATES. GPU00210
C IF (ZPP1.LT.0. .AND. ZPP2.LT.0.) RETURN GPU00220
C DELX=XPP2-XPP1 GPU00230
C DELY=YPP2-YPP1 GPU00240
C DELZ=ZPP2-ZPP1 GPU00250
C IF (ABS(DELX).LE..01*ABS(DELY) .OR. ABS(DELX).LE..01*ABS(DELY)) GPU00260
C 1 GOTO 10 GPU00270
C ICASE=1 GPU00280
C SX=DELX/DELX GPU00290
C SY=DELY/DELX GPU00300
C SZ=DELZ/DELX GPU00310
C XI=XPP1-SX*XPP1 GPU00320
C YI=YPP1-SY*XPP1 GPU00330
C ZI=ZPP1-SZ*XPP1 GPU00340
C AV(1)=XPP1 GPU00350
C AV(2)=XPP2 GPU00360
C GOTO 80 GPU00370
10 IF (ABS(DELX).LE..01*ABS(DELY)) GOTO 20 GPU00380
C ICASE=2 GPU00390
C SX=DELX/DELY GPU00400
C SY=DELY/DELY GPU00410
C SZ=DELZ/DELY GPU00420
C XI=XPP1-SX*YPP1 GPU00430
C YI=YPP1-SY*YPP1 GPU00440
C ZI=ZPP1-SZ*YPP1 GPU00450
C AV(1)=YPP1 GPU00460
C AV(2)=YPP2 GPU00470
C GOTO 80 GPU00480
20 IF (ABS(DELZ).LT..001) RETURN GPU00490
C ICASE=3 GPU00500
C SX=DELX/DELZ GPU00510
C SY=DELY/DELZ GPU00520
C SZ=DELZ/DELZ GPU00530
C XI=XPP1-SX*ZPP1 GPU00540
C YI=YPP1-SY*ZPP1 GPU00550
C ZI=ZPP1-SZ*ZPP1 GPU00560
C AV(1)=ZPP1 GPU00570
C AV(2)=ZPP2 GPU00580
C GOTO 80 GPU00590
80 IF (ZPP1.LT.0.) AV(1)=-ZI/SZ GPU00600
C IF (ZPP2.LT.0.) AV(2)=-ZI/SZ GPU00610
C SMUL=SQRT(SX*SX+SY*SY+SZ*SZ) GPU00620
C IF (ICASE.EQ.0) RETURN GPU00630
C*** COMPUTE GAUSSIAN PARAMETERS, REAL AND REFLECTED IMAGE CLOUD TO GPU00640
C ACCOUNT FOR GROUND REFLECTED SMOKE GPU00650
C SIGZ=(2.73+C2*A)/2.15 GPU00660
C ZB=C-(2.73+C2*A) GPU00670
C IF (ZB.GE.C) RETURN GPU00680
C SIGY=B*SQRT(1.-(ZB/C)**2)/2.15 GPU00690
C IF (ZB.LT.0.) SIGY=B/2.15 GPU00700

```

```

SIGX=SIGY
ASIG=(SX/SIGX)**2 + (SY/SIGY)**2 + (SZ/SIGZ)**2
BMEAN=2.*(SX*(XI-A)/SIGX**2) + (SY*YI/SIGY**2)
BP(1)=BMEAN+2.*SZ*(ZI-ZB)/SIGZ**2
BP(2)=BMEAN+2.*SZ*(ZI+ZB)/SIGZ**2
CTOT=(XI-A)/SIGX**2 + (YI/SIGY)**2
CP(1)=CTOT + ((ZI-ZB)/SIGZ)**2
CP(2)=CTOT + ((ZI+ZB)/SIGZ)**2
C*** CALCULATE FOR LOS INTEGRAL
CMUL=SMUL/(2.*3.14159*SIGX*SIGY*SIGZ*SQRT(ASIG))
DO 220 I=1,2
CEXU=.5*(CP(1)-(BP(1)**2)/(4.*ASIG))
IF (CEXU.GT.20.) GOTO 220
C*** INFINITE PATH LOS
CLU=EXP(-CEXU)
C*** CORRECTION FOR FINITE PATH
DO 210 J=1,2
AP1=(AV(J)+BP(1))/(2.*ASIG)*SQRT(ASIG/2.)
P1=ABS(AP1)
CP1=0.
IF (P1.LE.5.) CP1=0.5/(1.+P1*(.0705230784+P1*(.042282013+P1*(
*.0092705272+P1*(.0001520134+P1*(.0002765672+P1*(.0000430638))))))
**16
IF (AP1.GE.0.) CP1=1.-CP1
AP(J)=CP1
210 CONTINUE
CLGAUS=CLGAUS+CLU*ABS(AP(2)-AP(1))
220 CONTINUE
CLGAUS=CMUL*CLGAUS
RETURN
END

```

```

GPU00710
GPU00720
GPU00730
GPU00740
GPU00750
GPU00760
GPU00770
GPU00780
GPU00790
GPU00800
GPU00810
GPU00820
GPU00830
GPU00840
GPU00850
GPU00860
GPU00870
GPU00880
GPU00890
GPU00900
GPU00910
GPU00920
GPU00930
GPU00940
GPU00950
GPU00960
GPU00970
GPU00980
GPU00990
GPU01000
GPU01010

```

```

SUBROUTINE LZTRAN(WAVE1,ICLMAT,LAZTRN,IERR)
THIS SUBROUTINE CALCULATES MOLECULAR ABSORPTION COEFFICIENTS AT
LASER FREQUENCIES. PH20 AND T ARE THE PARTIAL PRESSURE OF WATER
VAPOR AND TEMPERATURE, IN TORR AND DEGREES K RESPECTIVELY.
LID IS THE LASER LINE IDENTIFICATION AS DESCRIBED BELOW.
ABCOEF IS THE ABSORPTION COEFFICIENT RETURNED IN KM-1.

*****
INPUT
CARD 1  LNAME1, LNAME2, PH20, T, LZPATH FORMAT (2A4,3F10.3)

LNAME1  FIRST 4 CHARS OF LASER LINE  (A4)
LNAME2  SECOND 4 CHARS OF LASER LINE  (A4)
      ***IF LNAME NOT ENTERED WILL USE WAVELENGTH READ IN MAIN

PH20    WATER VAPOR PRESSURE  0. TO .35  (MB)  (F10.3)
T       AMBIENT AIR TEMPERATURE 260. TO .320  (C)  (F10.3)

LZPATH  PATHLENGTH IN KM F(F10.3)
      ***PH20, AND T NOT REQUIRED WHEN ICLIMATE(IN MAIN)=1 OR 2
*****
OUTPUT
LZTRAN  TRANSMISSION

NOTES
ABCOEF  RETURNS THE ABSORPTION COEFFICIENT (KM-1)

LNAME1  FIRST 4 CHARS OF LASER LINE ON NORMAL RETURN
        BLANK ON ERROR RETURN

LNAME2  SECOND 4 CHARS OF LASER LINE ON NORMAL RETURN
        BLANK ON ERROR RETURN

++ CALLED PROGRAMS ++

LZIDNM

*****
LASER LINE IDENTIFICATION
LID=1  ND:YAG LASER, 1.06 MICRONS
LID=2  CO2 LASER LINE P(20) 10.591 MICRONS
LID=101 TO 127 OF LASER, 3.521 TO 4.089 MICRONS
  101 P3(12) * 107 P3(8) * 113 P3(5) * 119 P2(5) * 125 P1(5)
  102 P3(11) * 108 P2(11) * 114 P2(8) * 120 P1(8) * 126 P1(4)
  103 P3(10) * 109 P3(7) * 115 P2(7) * 121 P2(4) * 127 P1(3)
  104 P2(13) * 110 P2(10) * 116 P1(10) * 122 P1(7) *
  105 P3(9) * 111 P3(6) * 117 P2(6) * 123 P2(3) *
  106 P2(12) * 112 P2(9) * 118 P1(9) * 124 P1(6) *
LID=201 TO 219 CO LASER, 4.908 TO 5.088 MICRONS
  201 P6(12) * 205 P6(8) * 209 P5(12) * 213 P5(8) * 217 P4(9)
  202 P6(11) * 206 P5(15) * 210 P5(11) * 214 P5(7) * 218 P4(8)
  203 P6(10) * 207 P5(14) * 211 P5(10) * 215 P4(11) * 219 P4(7)
  204 P6(9) * 208 P5(13) * 212 P5(9) * 216 P4(10) *
LID=301 TO 305 GA AS LASER, (GA.85 TO GA.950) LASER LINE NAMES
  301 0.850 MICROMETERS * 304 0.925 MICROMETERS
  302 0.875 MICROMETERS * 305 0.950 MICROMETERS
  303 0.900 MICROMETERS *
  INTEGER LNAME1,LNAME2,BLANK,LNAME3,LNAME4
  REAL LAZTRN,LZPATH
  DIMENSION ADF0(30),ADF1(30),ADF2(30),ADF3(30),ADF4(30)
  1 ADF5(30)
  DIMENSION AC00(20),AC01(20),AC02(20),AC03(20),AC04(20)
  1 AC05(20)
  DIMENSION AGA0(5),AGA1(5),AGA2(5),AGA3(5),AGA4(5),
  1 AGA5(5),AGA6(5),AGA7(5),AGA8(5)
  COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK
  COMMON /CLYMAT/TEMP,PRESS,RH,AH,DP,VIS,CLDAMT,CLDHYT,
  1 FOGPRB,WINDVEL,WINDDIR,IPASCT

```

```

LZT00010
LZT00020
LZT00030
LZT00040
LZT00050
LZT00060
LZT00070
LZT00080
LZT00090
LZT00100
LZT00110
LZT00120
LZT00130
LZT00140
LZT00150
LZT00160
LZT00170
LZT00180
LZT00190
LZT00200
LZT00210
LZT00220
LZT00230
LZT00240
LZT00250
LZT00260
LZT00270
LZT00280
LZT00290
LZT00300
LZT00310
LZT00320
LZT00330
LZT00340
LZT00350
LZT00360
LZT00370
LZT00380
LZT00390
LZT00400
LZT00410
LZT00420
LZT00430
LZT00440
LZT00450
LZT00460
LZT00470
LZT00480
LZT00490
LZT00500
LZT00510
LZT00520
LZT00530
LZT00540
LZT00550
LZT00560
LZT00570
LZT00580
LZT00590
LZT00600
LZT00610
LZT00620
LZT00630
LZT00640
LZT00650
LZT00660
LZT00670
LZT00680
LZT00690
LZT00700

```

```

COMMON /IQUIN/IQIN, IQOUT, IPHFUN, LOUNIT, NDIRTU, NCLIMT, KSTOR, NPLLOTULZT00710
COMMON /GEOMET/PTS(15), IGEOSW
THE POLYNOMIAL COEFFICIENTS ARE SELECTED BY THE LID. THE
INDEX FOR THE COEFFICIENT ARRAYS FOR THE DF LASER IS
I = LID - 100. NOW I IS IN THE RANGE 1..27 SINCE THERE ARE
27 DF LASER LINES. WHEN THE POLYNOMIAL IS EVALUATED I IS
USED TO INDEX THE ARRAYS ADF/0..5/ THUS SELECTING THE
CORRECT COEFFICIENTS FOR THE LASER LINE SELECTED.
COEFFICIENTS FOR THE OTHER LASER POLYNOMIALS ARE SELECTED
IN THE SAME FASHION.
POLYNOMIAL COEFFICIENTS FOR DF LASER LINES, (1..27)
DATA ADF0/.1019,.08352,.04083,.03675,.02042,.01833,
.04738,.03134,.07870,.05844,.1096E-2,.9353E-
1 2,.2537E-2,.3254E-2,.1103E-2,.6471E-3,.1423E-
2 3,-.4664E-2,-.1221E-4,-.1698E-3,.1172E-3,.6195E-
3 42,.1272E-2,.5485E-2,.1651E-2,.6913E-2,-.4498E-
4 52,3*0./
DATA ADF1/-.9718E-4,-.1160E-3,-.4892E-4,-.4230E-4,-
1 2.1750E-4,-.1524E-4,-.5589E-4,-.4642E-4,-.1218E-
2 3,-.3683E-4,.1672E-4,-.1346E-4,-.4765E-6,-.8548E-
3 46,.6089E-5,.8897E-5,.1507E-5,.4448E-4,.4540E-
4 57,.5569E-6,.1327E-6,-.1462E-4,-.7524E-6,-.1398E-
5 67,.7025E-6,-.1044E-4,.1816E-4,3*0./
DATA ADF2/.9666E-2,.7252E-2,.7050E-2,.7142E-2,.6320E-
1 2.6191E-2,.5400E-2,.5344E-2,.4064E-2,.4682E-
2 3.3734E-2,.5839E-2,.5075E-2,.2484E-2,.8190E-
3 42,.6920E-2,.8779E-2,.7914E-2,.7094E-2,.5327E-
4 53,.6692E-2,.9452E-2,.01025,.01367,.01279,.7844E-
5 63,.01201,3*0./
DATA ADF3/-.2655E-4,-.1805E-4,-.1879E-4,-.1937E-4,-
1 2.1704E-4,-.1669E-4,-.1412E-4,-.1417E-4,-.9076E-
2 35,-.1165E-4,-.7371E-5,-.9113E-5,-.1290E-4,-
3 46,.1070E-5,-.6746E-5,-.1573E-4,-.1683E-4,-.1883E-
4 57,-.1774E-4,.3002E-4,-.1025E-4,-.2488E-4,-.2596E-
5 67,-.2848E-4,-.2858E-4,-.4991E-5,-.2608E-4,3*0./
DATA ADF4/.7847E-4,.5729E-4,.5585E-4,.5606E-4,.4847E-
1 2.4668E-4,.4145E-4,.4218E-4,.3314E-4,.3642E-
2 3.3170E-4,.3682E-4,.3835E-4,.4798E-4,.3537E-
3 44,.4635E-4,.4941E-4,.5601E-4,.5494E-4,-.1120E-
4 55,.6567E-4,.9104E-4,.7398E-4,.8509E-4,.8746E-
5 66,.1050E-3,.1060E-3,3*0./
DATA ADF5/-.2056E-6,-.1374E-6,-.1408E-6,-.1432E-6,-
1 2.1222E-6,-.1172E-6,-.1011E-6,-.1054E-6,-.7169E-
2 37,-.8600E-7,-.6588E-7,-.8998E-7,-.9081E-7,-
3 48,.1320E-6,-.1570E-7,-.9751E-7,-.8159E-7,-.1234E-
4 59,-.1313E-6,-.1836E-6,-.1469E-6,-.2540E-6,-.1796E-
5 69,-.1691E-6,-.1886E-6,-.2635E-6,-.2855E-6,3*0./
C POLYNOMIAL COEFFICIENTS FOR CO LASER LINES, (1..19)
DATA AC00/-1.813E-3,-9.289E-4,1.153E-3,-1.985E-3,-
1 2.4523E-3,-1.205E-3,-2.225E-4,-4.061E-3,-4.522E-
2 32,-2.267E-6,-5.917E-3,-1.423E-3,-3.640E-3,1.096E-
3 43,6.455E-4,-3.922E-3,-1.873E-5,-1.055E-4,1.489E-
4 54,2,0./
DATA AC01/3 426E-5,3.658E-6,-1.372E-6,7.229E-6,1.641E-
1 25,6.423E-6,1.042E-7,1.435E-5,1.593E-4,-5.334E-
2 36,1.498E-5,5.284E-6,1.806E-5,-3.651E-6,-1.303E-
3 47,1.835E-5,4.755E-6,2.330E-6,6.196E-6,0./
DATA AC02/8.813E-2,-1.020E-1,4.881E-2,6.872E-2,-
1 2.6244E-1,4.474E-2,1.226E-2,-1.462E-2,1.490E-
2 31,1.428E-2,1.934,9.034E-3,-1.091E-1,1.284E-2,-
3 42,1.131E-2,6.543E-3,2.824E-2,9.463E-3,-9.885E-2,0./
DATA AC03/8.384E-4,1.211E-3,-4.687E-6,-7.765E-
1 25,2.641E-3,-3.135E-5,1.620E-4,6.707E-4,2.581E-
2 36,3.4018E-4,-3.113E-3,9.692E-3,7.907E-4,1.070E-
3 47,3.026E-4,1.216E-4,-1.859E-5,4.274E-5,6.241E-
4 58,4,0./
DATA AC04/2.850E-4,-4.934E-5,-2.176E-5,1.253E-
1 25,1.158E-3,1.926E-5,9.823E-5,-2.395E-4,-6.255E-
2 36,5,-6.630E-5,4.851E-3,-5.183E-5,-6.993E-4,-8.196E-
3 47,5,-2.239E-4,-1.120E-4,5.415E-5,-8.393E-5,-5.893E-
4 58

```

```

4 DATA 4,0./
1 DATA AC05/4.209E-6,3.426E-6,5.383E-7,6.252E-8,-4.149E-
2 6,2.374E-7,-8.637E-9,3.454E-6,1.526E-5,2.140E-6,-
3 4.282E-6,5.433E-7,4.231E-6,7.778E-7,1.695E-
C POLYNOMIAL COEFFICIENTS FOR CO2 LASER
1 DATA AC020,AC021,AC022,AC023,AC024,AC025,AC026,AC027,
2 AC028/4.488,-4.1864E-3,5.7903E-2,-3.6770E-
3 4,3.8521E-3,-4.7330E-6,1.0000E-5,6.0131E-7,-
1.7441E-8/
C POLYNOMIAL COEFFICIENTS FOR GA AS LASER, (1..5)
1 DATA AGA0/7.947E-3,7.590E-3,1.010,0.6094,4.271/
2 DATA AGA1/-3.543E-5,-3.544E-5,-8.012E-3,-4.777E-3,-
3 3.371E-2/
1 DATA AGA2/1.740E-4,1.203E-4,8.736E-2,4.527E-2,.3364/
2 DATA AGA3/-5.855E-7,-5.093E-7,-2.135E-4,-1.104E-4,-
3 8.425E-4/
1 DATA AGA4/-1.282E-5,-9.236E-6,-3.964E-3,-2.154E-3,-
2 1.615E-2/
3 DATA AGA5/5.586E-8,4.720E-8,1.596E-5,8.512E-6,6.350E-
5/
1 DATA AGA6/5.124E-8,5.235E-8,1.600E-5,9.392E-6,6.705E-
5/
2 DATA AGA7/2.706E-10,4.379E-10,-1.396E-7,-6.982E-8,-
3 4.721E-7/
1 DATA AGA8/-4.963E-11,-5.728E-11,-1.125E-8,-5.657E-9,-
2 4.113E-8/
3 DATA BLANK/1H /
C CHANGE ACCURACY TO 3 DECIMAL PTS (PGM DATA LIMIT)
WAVEL=FLOAT(IFIX(1000.*WAVEL))/1000.
READ (IOIN,1100) LNAME1,LNAME3,LNAME2,LNAME4,PH20,T,LZPATH
IF(IGEOSW.NE.1)GO TO 99
LZPATH=SQRT((PTS(4)-PTS(1))*2+(PTS(5)-PTS(2))*2+
+(PTS(6)-PTS(3))*2)
99 CONTINUE
C CHANGE UNITS - MB TO TORR; C TO K
PH20=PH20/TORRMB
T=T+CDEGK
IF (ICLMAT.EQ.1) T=TEMP+CDEGK
IF (ICLMAT.EQ.1) PH20=6.11*10.**
1 (7.5*TEMP/(TEMP+237.3))*RH/(100.*TORRMB)
C PRINT HEADER WHEN THE WAVELENGTH CHANGES
IF (OLDWAV.NE.WAVEL) WRITE (IOOUT,1000)
OLDWAV=WAVEL
ABCOEF=0.
IF (WAVEL.EQ.0.0) GO TO 100
IF (WAVEL.LT.0.8.OR.WAVEL.GT.11.0) GO TO 900
100 CALL LZIDNM(WAVEL,LNAME1,LNAME3,LNAME2,LNAME4,LID)
IF (LID.EQ.0) RETURN
P2=PH20*PH20
IF (T.GE.260.AND.T.LE.320.AND.PH20.GE.0.AND.
1 PH20.LE.35) GO TO 200
C PRINT WARNING THAT TEMP OR PRESSURE IS OUT OF RANGE FOR
C ACCURATE CALCULATIONS AND CONTINUE.
WRITE (IOOUT,1300)
200 IF (LID.GT.100) GO TO 500
IF (LID.LT.1.OR.LID.GT.2) GO TO 900
IF (LID-2) 300,400,900
C ND:YAG LASER. NO MOLECULAR ABSORPTION AT 1.06 MICRONS.
300 GO TO 800
C CO2 LASER LINE P(20)
400 T2=T*T
ABCOEF=AC020+AC021*T+AC022*PH20+AC023*T*PH20+AC024*P2+
1 AC025*T*P2+AC026*T2+AC027*T2*PH20+AC028*T2*P2
GO TO 800
500 IF (LID.GT.200) GO TO 600
C DF LASER. I IS THE LASER LINE INDEX
I=LID-100
IF (I.GT.27) GO TO 900
ABCOEF=ADF0(I)+ADF1(I)*T+ADF2(I)*PH20+ADF3(I)*T*PH20+
LZT01410
LZT01420
LZT01430
LZT01440
LZT01450
LZT01460
LZT01470
LZT01480
LZT01490
LZT01500
LZT01510
LZT01520
LZT01530
LZT01540
LZT01550
LZT01560
LZT01570
LZT01580
LZT01590
LZT01600
LZT01610
LZT01620
LZT01630
LZT01640
LZT01650
LZT01660
LZT01670
LZT01680
LZT01690
LZT01700
LZT01710
LZT01720
LZT01730
LZT01740
LZT01750
LZT01760
LZT01770
LZT01780
LZT01790
LZT01800
LZT01810
LZT01820
LZT01830
LZT01840
LZT01850
LZT01860
LZT01870
LZT01880
LZT01890
LZT01900
LZT01910
LZT01920
LZT01930
LZT01940
LZT01950
LZT01960
LZT01970
LZT01980
LZT01990
LZT02000
LZT02010
LZT02020
LZT02030
LZT02040
LZT02050
LZT02060
LZT02070
LZT02080
LZT02090
LZT02100

```

```

      1      ADF4(I)*P2+ADF5(I)*T*P2
      GO TO 800
600 IF (LID.GT.300) GO TO 700
C CO LASER. I IS THE LASER LINE INDEX.
      I=LID-200
      IF (I.GT.19) GO TO 900
      ABCOEF=AC00(I)+AC01(I)*T+AC02(I)*PH20+AC03(I)*T*PH20+
      AC04(I)*P2+AC05(I)*T*P2
      GO TO 800
700 IF (LID.GT.400) GO TO 900
C GA AS LASER. I IS WAVELENGTH INDEX.
      I=LID-300
      IF (I.GT.5) GO TO 900
      T2=T*T
      ABCOEF=AGA0(I)+AGA1(I)*T+AGA2(I)*PH20+AGA3(I)*T*PH20+
      AGA4(I)*P2+AGA5(I)*T*P2+AGA6(I)*T2+AGA7(I)*T2*
      PH20+AGA8(I)*T2*P2
      GO TO 800
C NORMAL RETURN
800 IF (ABCOEF.LT.0.) ABCOEF=0.
C COMPUTE TRANSMISSION
      LAZTRN=EXP(-LZPATH*ABCOEF)
      WRITE (IOOUT,1200) WAVEL,PH20,T,ABCOEF,LNAME1,LNAME3,
      +LNAME2,LNAME4,LZPATH,LAZTRN
      RETURN
C ERROR RETURN
900 WRITE (IOOUT,1400)
      LNAME1=BLANK
      LNAME3=BLANK
      LNAME2=BLANK
      LNAME4=BLANK
      LAZTRN=1.
      IERR=1
      RETURN
C
1000 FORMAT (/ ,69X,10HABSORPTION,/ ,23X,11H WAVELENGTH,4X,
1      12HH2O PRESSURE,4X,11HTEMPERATURE,4X,
2      11HCOEFFICIENT,6X,4HLINE,9X,10HPATHLENGTH,4X,
3      12HTRANSMISSION,/ ,24X,9H(MICRONS),8X,6H(TORR)
4      ,11X,5H(ABS),10X,6H(KM-1),24X,4H(KM),/)
1100 FORMAT (4(A2),3F10.3)
1200 FORMAT (1H ,22X,F09.3,F15.3,F16.2,E16.3,7X,4(A2),5X,
1      E10.4,5X,E10.4)
1300 FORMAT (39H *** WARNING VALUE OF T OR PH20 OUT OF ,
1      10HRANGE *** /28H T RANGE = 260 TO 320 K ,
2      25HPH20 RANGE = 0 TO 35 TORR)
1400 FORMAT (40H *** ERROR WAVELENGTH OUT OF ACCEPTABLE
1      7HRANGE: ,26H .8 TO 11.0 MICRONS *** / ,
2      37H CONTROL RETURNED TO MAIN FROM LZTRAN)
      END

```

```

LZT02110
LZT02120
LZT02130
LZT02140
LZT02150
LZT02160
LZT02170
LZT02180
LZT02190
LZT02200
LZT02210
LZT02220
LZT02230
LZT02240
LZT02250
LZT02260
LZT02270
LZT02280
LZT02290
LZT02300
LZT02310
LZT02320
LZT02330
LZT02340
LZT02350
LZT02360
LZT02370
LZT02380
LZT02390
LZT02400
LZT02410
LZT02420
LZT02430
LZT02440
LZT02450
LZT02460
LZT02470
LZT02480
LZT02490
LZT02500
LZT02510
LZT02520
LZT02530
LZT02540
LZT02550
LZT02560
LZT02570
LZT02580

```



```

SUBROUTINE LZIDNM(WAVEL,LNAME1,LNAME3,LNAME2,LNAME4,LID)
*****
THIS SUBROUTINE CONVERTS THE WAVELENGTH IN MICRONS OR THE LASER
LINE NAME, IF WAVEL = 0, TO AN INTEGER LASER ID NUMBER WHICH IS
USED BY LZTRAN.
++ WAVEL TO LID, WHEN WAVEL NOT = 0 ++
THE SUBROUTINE DOES A BINARY SEARCH OF THE ARRAY AWAVEL TO FIND
A MATCH, WAVEL = AWAVEL(K). WHEN A MATCH IS FOUND THE LASER ID
IS LOADED FROM THE ID ARRAY, LID = AID(K). THE LASER LINE NAME
IS ALSO LOADED INTO TWO VARIABLES, LNAME1 = INAME1(K) LNAME2 =
INAME2(K). IF AN EXACT MATCH IS NOT FOUND THE CLOSEST STANDARD
TO THE PARAMETER WAVEL IS USED. A WARNING IS PRINTED AND K IS
SET TO THE PROPER VALUE SO IT CAN BE USED TO INDEX THE ID AND
LINE ARRAYS.
++ LASER LINE TO LID, WHEN WAVEL = 0 ++
WHEN THE WAVELENGTH PARAMETER IS ZERO THE CONVERSION IS DONE
FROM LASER LINE NAME TO LID. A SEQUENTIAL SEARCH OF THE LINE
NAME ARRAY IS PERFORMED. WHEN A MATCH IS FOUND K IS SET AND
WAVEL AND LID ARE LOADED FROM THE APPROPRIATE ARRAYS. WHEN
NO MATCH IS FOUND AN ERROR MESSAGE IS PRINTED AND LID IS SET
TO ZERO. LID IS USED TO NOTIFY LZTRAN THAT AN ERROR HAS
OCCURED AND NO CALCULATIONS SHOULD BE PERFORMED.

++ PARAMETERS ++
WAVEL    LASER WAVELENGTH (MICRONS)
*** INPUTS IF WAVEL = 0.0 ***
LNAME1   FIRST 4 CHARS OF LASER LINE
LNAME2   SECOND 4 CHARS OF LASER LINE
++ RESULTS ++
LID      LASER LINE IDENTIFIER
LNAME1   FIRST 4 CHARS OF LASER LINE
LNAME2   SECOND 4 CHARS OF LASER LINE
*****
INTEGER AID(53)
COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
DIMENSION AWAVEL(53),INAME1(106),INAME2(53)
DATA AWAVEL/.85,.875,.9,
1  925,.95,1.06,3.521,3.55,3.581,3.612,3.636,3.645,
2  3.666,3.679,3.698,3.715,3.731,3.752,3.765,3.8,
3  3.82,3.837,3.854,3.875,3.89,3.915,3.927,3.956,
4  3.965,3.999,4.005,4.046,4.089,4.908,4.918,4.928,
5  4.938,4.948,4.972,4.982,4.992,5.002,5.012,5.022,
6  5.032,5.043,5.047,5.054,5.057,5.067,5.078,5.088,
7  10.591/
DATA AID/301,302,303,304,305,1,127,126,125,124,123,
1  122,121,120,119,118,117,116,115,114,113,112,111,
2  110,109,108,107,106,105,104,103,102,101,219,218,
3  217,216,215,214,213,212,211,210,209,208,207,205,
4  206,204,203,202,201,2/
DATA INAME1/2HGA,2H.8,2HGA,2H.8,2HGA,2H.9,2HGA,2H.9,2HGA,2H.9,
1  2HRU,2HBY,2HP1,2H(3,2HP1,2H(4,2HP1,2H(5,2HP1,2H(6,2HP2,2H(3,
2  2HP1,2H(7,2HP2,2H(4,2HP1,2H(8,2HP2,2H(5,2HP1,2H(9,2HP2,2H(6,
3  2HP1,2H(1,2HP2,2H(7,2HP2,2H(8,2HP3,2H(5,2HP2,2H(9,2HP3,2H(6,
4  2HP2,2H(1,2HP3,2H(7,2HP2,2H(1,2HP3,2H(8,2HP2,2H(1,2HP3,2H(9,
5  2HP2,2H(1,2HP3,2H(1,2HP3,2H(1,2HP3,2H(1,2HP4,2H(7,2HP4,2H(8,
6  2HP4,2H(9,2HP4,2H(1,2HP4,2H(1,2HP5,2H(7,2HP5,2H(8,2HP5,2H(9,
7  2HP5,2H(1,2HP5,2H(1,2HP5,2H(1,2HP5,2H(1,2HP5,2H(1,2HP6,2H(8,
8  2HP5,2H(1,2HP6,2H(9,2HP6,2H(1,2HP6,2H(1,2HP6,2H(1,2HP(,2H20/
DATA INAME2/1H5,2H75,1H,2H25,1H5,1H,11*1H,2H0,5*
1  1H,2H0,1H,2H1,1H,2H2,1H,2H3,2H0,2H1,
2  2H2,3*1H,2H0,2H1,3*1H,2H0,2H1,2H2,2H3,

```

```

3      2H4),1H),2H5),1H),2H0),2H1),2H2),1H)/
DATA IFIRST,ILAST/1,53/
I=IFIRST
J=ILAST
TWAVEL=WAVEL
C CHECK FOR WAVELENGTH OR LASER LINE PASSED AS INPUT. IF
C WAVEL = 0 THEN DO A SEQUENTIAL SEARCH ON LINE NAME,
C LNAME1,LNAME2,
IF (WAVEL.NE.0.0) GO TO 200
C INPUT = LASER LINE NAME
C SEQUENTIAL SEARCH LOOP
DO 100 K=IFIRST,ILAST
KK=2*K-1
IF (LNAME1.EQ.INAME1(KK)).AND.(LNAME3.EQ.INAME1(KK+1))) GO TO 10
GO TO 100
10 IF (LNAME2.EQ.INAME2(K)) GO TO 600
100 CONTINUE
C ERROR, NO MATCH ON LASER NAME
LID=0
C PRINT ERROR MESSAGE
WRITE (1000,900) LNAME1,LNAME3,LNAME2,LNAME4
GO TO 700
C INPUT = WAVELENGTH
C BINARY SEARCH LOOP
200 K=(I+J)/2
IF (WAVEL.LE.AWAVEL(K)) J=K-1
IF (WAVEL.GE.AWAVEL(K)) I=K+1
IF (I.LE.J) GO TO 200
C DID WAVELENGTH MATCH A STANDARD IN AWAVEL(*) ?
IF (I-1.GT.J) GO TO 600
C WAVELENGTH NOT EXACTLY EQUAL TO ONE OF THE STANDARDS IN ARRAY
C AWAVEL. CHANGE WAVELENGTH TO EQUAL THE STANDARD IT IS CLOSEST TO.
C THEN PRINT WARNING OF CHANGE.
IF (WAVEL.GT.AWAVEL(K).AND.K.EQ.ILAST) GO TO 500
IF (WAVEL.LT.AWAVEL(K).AND.K.EQ.IFIRST) GO TO 500
IF (WAVEL-AWAVEL(K)) 300,600,400
C WAVEL LT AWAVEL(K)
C CHECK IF CLOSER TO AWAVEL(K) OR AWAVEL(K-1)
300 DELTA1=WAVEL-AWAVEL(K-1)
DELTA2=AWAVEL(K)-WAVEL
IF (DELTA1.LT.DELTA2) K=K-1
GO TO 500
C WAVEL GT AWAVEL(K)
C CHECK IF CLOSER TO AWAVEL(K) OR AWAVEL(K+1)
400 DELTA1=WAVEL-AWAVEL(K)
DELTA2=AWAVEL(K+1)-WAVEL
IF (DELTA2.LT.DELTA1) K=K+1
C PRINT WARNING
500 WAVEL=AWAVEL(K)
WRITE (1000,800) TWAVEL,WAVEL
C LOAD LASER ID NUMBER
600 LID=AID(K)
C LOAD LINE NAME
KK=2*K-1
LNAME1=INAME1(KK)
LNAME3=INAME1(KK+1)
LNAME2=INAME2(K)
C LOAD WAVELENGTH
WAVEL=AWAVEL(K)
700 RETURN
C
800 FORMAT (29H *** WARNING INPUT WAVELENGTH,F7.3,
1      11H CHANGED TO,F7.3,10H NEAREST STANDARD ,
2      14HWAVELENGTH ***)
900 FORMAT (24H *** ERROR LASER LINE #,4(A2),6H# NOT ,
1      9HVALID ***,/,26H CONTROL RETURNED TO MAIN ,
2      12HFROM LZTRAN.)
END

```

```

LZI00690
LZI00700
LZI00710
LZI00720
LZI00730
LZI00740
LZI00750
LZI00760
LZI00770
LZI00780
LZI00790
LZI00800
LZI00820
LZI00830
LZI00840
LZI00850
LZI00860
LZI00870
LZI00880
LZI00890
LZI00900
LZI00910
LZI00920
LZI00930
LZI00940
LZI00950
LZI00960
LZI00970
LZI00980
LZI00990
LZI01000
LZI01010
LZI01020
LZI01030
LZI01040
LZI01050
LZI01060
LZI01070
LZI01080
LZI01090
LZI01100
LZI01110
LZI01120
LZI01130
LZI01140
LZI01150
LZI01160
LZI01170
LZI01180
LZI01190
LZI01200
LZI01210
LZI01220
LZI01230
LZI01240
LZI01250
LZI01260
LZI01270
LZI01280
LZI01290
LZI01300
LZI01310
LZI01320

```

SUBROUTINE DRTRAN(WAVE1,ICLMAT,TRNLOS,IERR)

PURPOSE

DIRTRAN-2 EXPLOSION PRODUCED AND VEHICLE GENERATED DUST MODEL

INPUT,OUTPUT AND CALLING PROGRAM

INPUTS

VALUES IN ARGUMENT LIST

ICLMAT INTEGER VALUE USED TO INDICATE HOW METEOROLOGICAL DATA IS TO BE MADE AVAILABLE IF ICLMAT IS
 0 - MET1 IDENTIFIER WITH THE APPROPRIATE PARAMETERS ARE TO BE READ IN
 1 - NECESSARY METEOROLOGICAL DATA IS PASSED IN COMMON/CLYMAT/ AND MET1 IS NOT TO BE READ IN

WAVE1 WAVELENGTH IN MICROMETERS. USED TO DETERMINE NWL, WHERE NWL IS AN INTEGER INDEX FOR WAVELENGTH DETERMINED WITHIN THE CODE

NWL	WAVE1
1	FOR 0.4 - 0.7 MICROMETER (VISIBLE)
2	FOR 0.8 - 1.1 MICROMETER
3	FOR 3.5 - 4.0 MICROMETER
4	FOR 8.5 - 12.0 MICROMETER
5	FOR 2100 - 3200 MICROMETER

INPUTS TO BE READ

EACH INPUT RECORD BEGINS WITH A 4 LETTER IDENTIFIER IN COLUMNS 1-4 FOLLOWED BY AS MANY (REAL) FIELDS AS NEEDED, UP TO 9, 8 COLUMNS PER FIELD BEGINNING IN COLUMN 9.
 THE INPUT FILE MAY CONTAIN SEVERAL SEQUENCES OF THE FOLLOWING RECORDS. EACH SEQUENCE SEPERATED BY A GO CARD. ONCE THE INITIAL SEQUENCE HAS BEEN READ IN AND THE MINIMUM REQUIREMENTS FOR EXECUTION OF THE DESIRED OPTION HAS BEEN SATISFIED, ANY FOLLOWING SEQUENCE MAY CONTAIN A SUBSET OF THE INITIAL RECORDS REDEFINING INPUT VARIABLES AS DESIRED OR MAY CONTAIN A COMPLETELY NEW SET OF RECORDS.

** EACH SET OF INPUTS MUST END WITH A DONE CARD**

RECORD 1 MET1

NATMOS INTEGER WITH VALUES 1 TO 6 CORRESPONDING TO PASQUILL CATEGORIES A TO F.

ZTMP THE HEIGHT AT WHICH A TEMPERATURE MEASUREMENT IS AVAILABLE. VALID RANGE 0.5 - 100.0 M.

TMPMES THE TEMPERATURE MEASURED IN DEGREES KELVIN TAKEN AT HEIGHT ZTMP. VALID RANGE 270.0 - 315.0.

ZWND THE HEIGHT AT WHICH A WIND SPEED MEASUREMENT IS AVAILABLE. VALID RANGE 0.5 - 100.0 M.

WNMES THE WIND SPEED IN METERS/SECOND MEASURED AT ZWND VALID RANGE .1 - 20.0 M/S

THWND THE ANGLE THAT THE WIND VELOCITY VECTOR MAKES WITH THE USER'S POSITIVE X AXIS MEASURED IN DEGREES COUNTERCLOCKWISE, WHERE THE USERS POSITIVE X-AXIS POINTS EAST. THUS THWND IS THE ANGLE THAT THE WIND VELOCITY VECTOR MAKES WITH THE EAST.
 VALID RANGE: -360.0 - 360.0 DEGREES.
 NOTE: THWND IS NOT NEEDED FOR OPTION 3

RECORD 2		DRT00700
MET2		DRT00710
ID	A FLAG TO INDICATE WHETHER THE INVERSION LAYER HEIGHT IS GROWING OR NOT. IF ID IS	DRT00720
	0. THE INVERSION LAYER HEIGHT IS RELATIVELY CONSTANT	DRT00730
	1. THE INVERSION LAYER HEIGHT IS GROWING	DRT00740
PHI	THE LATITUDE OF THE DETONATION SITE,	DRT00750
	VALID RANGE 1.0 - 90. DEGREES. THAT IS THE NORTHERN	DRT00760
	HEMISPHERE.	DRT00770
RECORD 3		DRT00780
SOIL		DRT00790
NSOIL	INTEGER INDEX OF SOIL TYPE. NSOIL IS	DRT00800
	1. FOR SOIL-1, (DATA GRAF-II) EXPLOSIONS ONLY.	DRT00810
	2. FOR SOIL-2, (DATA DIRT-I) EXPLOSIONS ONLY.	DRT00820
	3. FOR SOIL-3, (DATA SMOKEWEEK-II) VEHICLES ONLY.	DRT00830
DSOD	DEPTH OF SOD IN METERS	DRT00840
	VALID RANGE: 0.0 - 1.0 M.	DRT00850
	NOTE: FOR VEHICLE MODEL IF DSOD>0.0 NO DUST IS	DRT00860
	GENERATED	DRT00870
SILT	SILT CONTENT OF SOIL (PARTICLE DIAMETERS < 75 MICRONS)	DRT00880
	I.E. SILT=.15 INDICATES A SILT CONTENT OF 15%	DRT00890
	NOTE: THIS INPUT NEEDED ONLY FOR VEHICLE MODEL (IOPT=5)	DRT00900
RECORD 4		DRT00910
CHAR		DRT00920
NCHRG	CHARGE TYPE INDEX WITH FOLLOWING VALUES	DRT00930
	1. SURFACE - LIVE FIRE OR 30 DEGREE TILTED	DRT00940
	STATIC, TIP ON GROUND	DRT00950
	2. BARE CHARGE ON SURFACE	DRT00960
	3. 30 DEGREE TILTED TIP AT 0.3 METER DEPTH	DRT00970
	4. 30 DEGREE TILTED TIP AT 0.6 METER DEPTH	DRT00980
	5. HORIZONTAL PROJECTILE ON SURFACE	DRT00990
	DEFAULT VALUE IS 1 IF NCHRG IS NOT BETWEEN 1 AND 5.	DRT01000
CHWT	THE WEIGHT OF THE CHARGE IN KG-TNT.	DRT01010
	VALID RANGE: 0.1 - 100.0 KG-TNT.	DRT01020
DETDEP	THE DEPTH OF DETONATION IN METERS.	DRT01030
	VALID RANGE: 0.0 - 2.0 M.	DRT01040
RECORD 5		DRT01050
EXPL		DRT01060
NARY	TYPE OF CHARGE DISTRIBUTION (USED FOR PROPER INPUT AND	DRT01070
	OUTPUT FORMATS) IF THE VALUE OF NARY AND IOPT OF THE GO	DRT01080
	CARD ARE NOT COMPATIBLE CATASTROPHE COULD RESULT!	DRT01090
	NOTE: WHEN NARY IS	DRT01100
	1. IOPT MUST ALSO BE 1.	DRT01110
	2. IOPT MUST ALSO BE 2.	DRT01120
	3. IOPT MUST BE 4.	DRT01130
	1.-SIMULTANEOUS BURST, UNIFORMLY DISTRIBUTED CHARGES	DRT01140
	IN A PARALLELOGRAM.	DRT01150
	(SPECIAL CASES ARE ,SINGLE CHARGE ,RECTANGLE AND	DRT01160
	ZIG ZAG PATTERN)	DRT01170
	2.-SIMULTANEOUS BURST, RANDOMLY DISTRIBUTED CHARGES.	DRT01180
	3.-SEQUENTIAL IN TIME AND RANDOM IN SPACE DISTRIBUTION OF	DRT01190
	CHARGES.	DRT01200
	NOTE:	DRT01210
	WHEN NARY=2. EACH CHARGE LOCATION MUST BE SPECIFIED	DRT01220
		DRT01230
		DRT01240
		DRT01250
		DRT01260
		DRT01270
		DRT01280
		DRT01290
		DRT01300
		DRT01310
		DRT01320
		DRT01330
		DRT01340
		DRT01350
		DRT01360
		DRT01370
		DRT01380
		DRT01390

WHEN NARY=3, EACH CHARGE LOCATION AND DETONATION TIME MUST BE SPECIFIED. THE CHARGE LOCATIONS ARE INPUT DIRECTLY FOLLOWING THIS INPUT RECORD WITH ONE LOCATION (AND DETONATION TIME IF APPROPRIATE) PER RECORD.

NCHS SINGLY DIMENSIONED ARRAY SPECIFYING NUMBER OF CHARGES WITH THE MAXIMUM TOTAL OF CHARGES 200. WHEN NARY=1, NCHS(1) IS THE NUMBER OF CHARGES IN THE DIRECTION OF SIDE1 AND NCHS(2) IS THE NUMBER OF CHARGES IN THE DIRECTION OF SIDE2. FOR A SINGLE CHARGE SET NCHS(1)=NCHS(2)=1. WHEN NARY=2, OR 3, NCHS(1) IS THE TOTAL NUMBER OF CHARGES AND SET NCHS(2)=1..

SRCBAS SINGLY DIMENSIONED ARRAY CONTAINING THE COORDINATES OF A CORNER POINT OF THE BOUNDING PARALLELOGRAM WHEN NARY=1. AND IS ALSO USED AS THE REFERENCE CHARGE BY THE OBSERVER. THAT IS, SRCBAS IS THE ORIGIN OF THE OBSERVERS COORDINATE SYSTEM. WHEN NARY=2, OR 3, SRCBAS SHOULD NOT APPEAR ON THE INPUT FILE AS COOR(I,1) IS USED AS THE REFERENCE CHARGE. WHERE COOR(I,1) IS THE FIRST CHARGE LOCATION SPECIFIED. VALID RANGE -10000.0 - 10000.0

SIDE1 SINGLY DIMENSIONED ARRAY NECESSARY ONLY WHEN NARY=1., SPECIFYING ONE SIDE OF THE BOUNDING PARALLELOGRAM FROM THE POINT SRCBAS(I). THAT IS, SIDE1 IS A VECTOR TO THE NEXT CHARGE ALONG ONE SIDE OF THE PARALLELOGRAM. WHEN NARY=2, OR 3, THIS VARIABLE SHOULD NOT APPEAR ON THE INPUT FILE.

SIDE2 SINGLY DIMENSIONED ARRAY NECESSARY ONLY WHEN NARY=1., SPECIFYING A SECOND SIDE OF THE BOUNDING PARALLELOGRAM FROM THE POINT SRCBAS(I). THAT IS SIDE2 IS A VECTOR TO THE NEXT CHARGE ALONG THE SECOND SIDE OF THE PARALLELOGRAM. WHEN NARY=2, OR 3, THIS VARIABLE SHOULD NOT APPEAR ON THE INPUT FILE.

** THE FOLLOWING RECORD MUST APPEAR THE APPROPRIATE NUMBER OF TIMES
 ** IMMEDIATELY FOLLOWING THE ABOVE RECORD IF NARY IS 2, OR 3, ON THE
 ** ABOVE RECORD. THAT IS IT MUST APPEAR THE SAME NUMBER OF TIMES
 ** AS THERE ARE CHARGES AS SPECIFIED ON RECORD EXPL. (I.E. IF NARY=2,
 ** AND NCHS=5, THE THIS RECORD MUST APPEAR 5 TIMES.

RECORD 6
 LUCA

COOR DOUBLY DIMENSIONED ARRAY CONTAINING THE DETONATION COORDINATES FOR EACH CHARGE WHEN NARY=2, OR 3.. WHEN NARY=1, THIS VARIABLE NEED NOT BE SPECIFIED AS THE CHARGE LOCATIONS ARE CALCULATED IN THE CODE FROM NCHS, SIDE1, SIDE2.

TSTAG SINGLY DIMENSIONED ARRAY CONTAINING THE TIME OF DETONATION OF EACH CHARGE. THIS IS ONLY SPECIFIED WHEN NARY=3..

RECORD 7
 VEH

V0 DOUBLY DIMENSIONED ARRAY CONTAINING THE INITIAL POSITION OF THE VEHICLE. V0(1)=X-COORDINATE V0(2)=Y-COORDINATE. VALID RANGE: -10000.0 - 10000.0

VEHDIR VEHICLE DIRECTION THE ANGLE THAT THE VEHICLE VELOCITY VECTOR MAKES WITH THE USER'S POSITIVE X-AXIS MEASURED IN DEGREES COUNTERCLOCKWISE. WHERE THE USER'S POSITIVE

X-AXIS POINTS EAST, THUS VEHDIR IS THE ANGLE THE
VELOCITY VECTOR MAKES WITH THE EAST.
VALID RANGE: -360.0 - 360.0

VEHSPD VEHICLE SPEED IN M/S
VEHWID VEHICLE WIDTH IN METERS
VEHWHT VEHICLE WEIGHT IN KGS.
VEHTYP TRACTION MECHANISM
 =0. VEHICLE HAS TIRES
 =1. VEHICLE IS TRACKED

RECORD 8
TRNC

TRNCOR A SINGLE DIMENSIONED ARRAY CONTAINING THE THREE
COORDINATES OF THE TRANSMITTER. THE COORDINATE SYSTEM
MUST BE IN METERS. THE THIRD COORDINATE IS RESTRICTED
TO BE BETWEEN .5 AND 10000.0 METERS (HEIGHT).
VALID RANGE OF THE FIRST TWO COORDINATES:
-10000.0 - 10000.0 M.

** IF THE COORDINATES ARE PASSED THROUGH THE GEOMET OPTION, THEN THE
** ARRAY TRNCOR NEED NOT BE SPECIFIED.

TRNMIN VALUE SUCH THAT A TRANSMITTANCE BELOW THIS VALUE CAN
BE CONSIDERED ZERO. DEFAULT IS 1.E-05
VALID RANGE: 1.0 - 1.E-05

RECORD 9
RECC

RECCOR A SINGLE DIMENSIONED ARRAY CONTAINING THE THREE
COORDINATES OF THE RECEIVER. (METERS)
THE THIRD COORDINATE IS RESTRICTED TO BE BETWEEN
.5 AND 10000.0 METERS. VALID RANGE OF THE FIRST TWO
COORDINATES IS: -10000.0 - 10000.0 M.

** IF THE COORDINATES ARE PASSED THROUGH THE GEOMET OPTION, THEN THE
** ARRAY RECCOR NEED NOT BE SPECIFIED.

RECORD 10
OBSC

OBSCOR A SINGLE DIMENSIONED ARRAY CONTAINING THE X AND Y
COORDINATES, RESP., OF THE OBSERVER. (METERS)
VALID RANGE: -10000.0 - 10000.0

SPCHT A SPECIFIED HEIGHT IN METERS AT WHICH THE WIDTH OF
THE CLOUD AS VIEWED FROM POSITION OBSCOR IS DESIRED.
MUST BE BETWEEN 1. AND 5. METERS.

** IF THE COORDINATES ARE PASSED THROUGH THE GEOMET OPTION, THEN THE
** ARRAY OBSCOR AND VARIABLE SPCHT NEED NOT BE SPECIFIED.

RECORD 11
TIMS

TSTART TIME AFTER DETONATION TO START TRANSMITTANCE AND/OR
CLOUD DIMENSION CALCULATIONS
VALID RANGE: .5 - 1000.0 SEC.

TEND TIME AFTER DETONATION TO TERMINATE TRANSMITTANCE
AND/OR CLOUD DIMENSIONS.
VALID RANGE: .5-1000.0 SEC. (TEND MUST BE .GE. TSTART)

TINC TIME INCREMENT BETWEEN CALCULATIONS

DRT02100
DRT02110
DRT02120
DRT02130
DRT02140
DRT02150
DRT02160
DRT02170
DRT02180
DRT02190
DRT02200
DRT02210
DRT02220
DRT02230
DRT02240
DRT02250
DRT02260
DRT02270
DRT02280
DRT02290
DRT02300
DRT02310
DRT02320
DRT02330

DRT02340
DRT02350
DRT02360
DRT02370
DRT02380
DRT02390
DRT02400
DRT02410
DRT02420
DRT02430
DRT02440
DRT02450
DRT02460

DRT02470
DRT02480
DRT02490
DRT02500
DRT02510
DRT02520
DRT02530
DRT02540
DRT02550
DRT02560
DRT02570

DRT02580
DRT02590
DRT02600
DRT02610
DRT02620
DRT02630
DRT02640
DRT02650
DRT02660
DRT02670
DRT02680
DRT02690
DRT02700

```

RECORD 12
GO ** THIS CARD INDICATES THAT THIS SEQUENCE OF INPUTS ARE
      COMPLETE AND CALCULATIONS ARE TO BEGIN.

      IOPT  OPTION TO BE USED

            1. SIMULTANEOUS BURST ,UNIFORMLY DISTRIBUTED CHARGES IN A
              PARALLELOGRAM

            2. SIMUTANEOUS BURST, RANDOMLY DISTRIBUTED CHARGES

            3. THE CODE IS TO PRECOMPUTE A SINGLE CLOUD AND STORE ON
              AN EXTERNAL FILE FOR USE LATER

            4. THE CODE IS TO USE A CLOUD THAT HAS BEEN PRECOMPUTED
              (NO CLOUD DIMENSIONS ARE COMPUTED FOR THIS OPTION)

            5. VEHICLE DUST MODEL

      IFILE  FORTRAN LOGICAL UNIT TO WHICH THE CODE IS TO WRITE FOR
            OPTION 3 OTHERWISE IT NEED NOT BE SPECIFIED

RECORD 13
DONE ** THIS RECORD INDICATES THAT THE USER HAS COMPLETED HIS
      DESIRED SEQUENCE OF INPUTS AND ALL CALCULATIONS ARE
      TERMINATED

      OUTPUTS

      ZINV   THE ESTIMATED INVERSION HEIGHT.

      TRNLOS THE TRANSMITTANCE ALONG THE LINE OF SIGHT BETWEEN
            THE TRANSMITTER AND THE RECEIVER.

      IERR   INTEGER ERROR CODE WHICH EQUALS 1 IF A FATAL ERROR
            OCCURS AND 0 OTHERWISE

      NERR   INTEGER ERROR CODE WITH THE VALUES
            0 NO ERRORS
            4 NO TRANSMITTER AND RECEIVER OR OBSERVER
              COORDINATES WERE SPECIFIED SO NO RESULTS WERE
              CALCULATED.
            7 THE CALCULATION OF ATMOSPHERIC PARAMETERS DID
              NOT CONVERGE.

      CNTRD  A SINGLY DIMENSIONED ARRAY CONTAINING THE HORIZONTAL
            COORDINATE AND THE VERTICAL COORDINATE OF THE
            CENTROID OF THE CLOUD.

      HEIGHT THE HEIGHT OF THE CLOUD IN METERS.

      CENWTH THE WIDTH OF THE CLOUD IN METERS AT THE CENTROID
            HEIGHT

      SPCWTH THE WIDTH OF THE CLOUD IN METERS AT THE SPECIFIED
            HEIGHT

      NCPTS  THE NUMBER OF POINTS DETERMINED ON THE EDGE OF THE
            CLOUD.

      CPTS   A DOUBLY DIMENSIONED ARRAY CONTAINING THE COORDINATES
            OF POINTS ON THE EDGE OF THE CLOUD. CPTS(I,J)
            IS THE HORIZONTAL COORDINATE OF THE J-TH POINT
            AND CPTS(2,J)IS THE VERTICAL COORDINATE OF THE
            J-TH POINT. THE FIRST INDEX MUST BE DIMENSIONED
            TO 2.

```

DRT02710
DRT02720
DRT02730
DRT02740
DRT02750
DRT02760
DRT02770
DRT02780
DRT02790
DRT02800
DRT02810
DRT02820
DRT02830
DRT02840
DRT02850
DRT02860
DRT02870
DRT02880
DRT02890
DRT02900
DRT02910
DRT02920
DRT02930
DRT02940
DRT02950
DRT02960
DRT02970
DRT02980
DRT02990
DRT03000
DRT03010
DRT03020
DRT03030
DRT03040
DRT03050
DRT03060
DRT03070
DRT03080
DRT03090
DRT03100
DRT03110
DRT03120
DRT03130
DRT03140
DRT03150
DRT03160
DRT03170
DRT03180
DRT03190
DRT03200
DRT03210
DRT03220
DRT03230
DRT03240
DRT03250
DRT03260
DRT03270
DRT03280
DRT03290
DRT03300
DRT03310
DRT03320
DRT03330
DRT03340
DRT03350
DRT03360
DRT03370
DRT03380
DRT03390
DRT03400

```

C SUBROUTINES CALLED
C
C DUSTCL CONTROLLING ROUTINE FOR THE CALCULATION OF CLOUD DIMENSIONS
C AND TRANSMITTANCES THROUGH DUST CLOUDS FOR OPTIONS 1 AND 2
C GIVEN METEOROLOGICAL DATA, SOIL AND EXPLOSIVE
C CHARACTERISTICS, AND WAVELENGTH.
C
C COMPCL CONTROLLING ROUTINE FOR PRECOMPUTING A SINGLE EXPLOSION
C (OPTION 3) GIVEN METEOROLOGICAL DATA, SOIL AND EXPLOSION
C CHARACTERISTICS. ALSO USES THIS PRECOMPUTED CLOUD AT SOME
C LATER RUNNING OF THE CODE (OPTION 4) TO ESTIMATE A
C TRANSMITTANCE GIVEN TRANSMITTER AND RECEIVER COORDINATES.
C
C VEHCL CONTROLLING ROUTINE FOR THE CALCULATION OF A TRANSMITTANCE
C THROUGH A VEHICLE GENERATED DUST CLOUD (OPTION 5) GIVEN
C METEOROLOGICAL DATA, SOIL CHARACTERISTICS, VEHICLE
C CHARACTERISTICS, AND WAVELENGTH.
C
C *****
C LOGICAL NEWATM,NEWSRC,LOSTRN,EDGE,NEWTIM, CLMRED,DHDT,ONCE
C LOGICAL TEST,NEWVEH,NEWCOR
C LOGICAL M1,M2,SL,CH,EX,TC,RC,OC,TM,VH
C INTEGER VEHTYP
C REAL M,N
C DIMENSION ZTMP(2),TMPMES(2),ZWND(2),WDMES(2),TRNCOR(3)
C DIMENSION SRCBAS(2),SIDE1(2),SIDE2(2),NCHS(2)
C 1,RECCOR(3),CPTS(2,6),CNTRD(2),OBSCOR(2)
C DIMENSION RDIN(10),RKEY(12),V0(2),PAS(6)
C COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
C COMMON /CLYMAT/ TEMP,PRESS,RH,AH,DP,VIS,CLDAMT,CLDHYT,
C 1 FOGPRB,WNDVEL,WINDIR,IPASCT
C COMMON/MOS/DIFF(2,200),NCHTOT,PRSEP(200),NTOT,NARY,ITOT,
C + COOR(2,200),TSTAG(200),DMMY(401)
C COMMON/WNDPRM/DXZ0,DYX0,DZ0,U0,M,N,ZINV
C COMMON/TRANNY/THRESH,TEST,NWL,NSOIL
C COMMON/GEOM/COSTH2,SINTH,SINTH2,VISEXT,RTPI,SCRN(2)
C COMMON /GEOMET/PTS(15),IGEOSW
C COMMON/OPTION/IOPT,IFILE
C DATA RKEY/4HMET1,4HMET2,4HISOIL,4HCHAR,4HEXPL,4HVEHC,4HTRNC,
C 1 4HRECC,4HOBSC,4HTIMS,4HGO,4HDONE,
C DATA PAS/4HA,4HB,4HC,4HD,4HE,4HF
C DATA M1,M2,SL,CH,EX,TC,RC,OC,TM,VH/.FALSE.,.FALSE.,.FALSE.,
C 1 .FALSE.,.FALSE.,.FALSE.,.FALSE.,.FALSE.,.FALSE.,.FALSE./
C DATA NEWATM,NEWSRC,NEWVEH,LOSTRN,EDGE,NEWTIM/.FALSE.,.FALSE.,
C 1 .FALSE.,.FALSE.,.FALSE.,.FALSE./
C DATA NEWCOR/.FALSE./
C DATA VISEXT,RTPI/.1,1.772454/
C IERR=0
C CLMRED=.FALSE.
C DHDT=.FALSE.
C ONCE=.FALSE.
C TEST=.FALSE.
C WRITE(100UT,800)
C 800 FORMAT(1H0,36X,42HDIRTRAN-2 DUST CLOUD INFRARED TRANSMISSION,
C 1 15H CALCULATION,/,36X,60H*** NOTE -- ALL UNITS ARE MKS UNLESS
C 2 OTHERWISE SPECIFIED **/,/)
C DO 5 K=1,200
C TSTAG(K)=0.0
C 5 CONTINUE
C
C DETERMINE INTEGER INDEX FOR WAVELENGTH
C
C 10 IF(WAVE1.LT.0.4)GO TO 29
C IF(WAVE1.GT.0.7)GO TO 21
C NWL=1
C GO TO 30
C 21 IF(WAVE1.LT.0.8)GO TO 29
C IF(WAVE1.GT.1.1)GO TO 22
C NWL=2
C GO TO 30

```


22	IF(WAVE1.LT.3.5)GO TO 29	DRT04100
	IF(WAVE1.GT.4.0)GO TO 23	DRT04110
	NWL=3	DRT04120
	GO TO 30	DRT04130
23	IF(WAVE1.LT.8.5)GO TO 29	DRT04140
	IF(WAVE1.GT.12.0)GO TO 24	DRT04150
	NWL=4	DRT04160
	GO TO 30	DRT04170
24	IF(WAVE1.LT.2100.)GO TO 29	DRT04180
	IF(WAVE1.GT.3200.)GO TO 29	DRT04190
	NWL=5	DRT04200
	GO TO 30	DRT04210
29	WRITE(10OUT,802)	DRT04220
802	FORMAT(37X,38H*** DIRTRAN ERROR - WAVE1 OUT OF RANGE)	DRT04230
	IERR=1	DRT04240
	GO TO 999	DRT04250
30	CONTINUE	DRT04260
C	READ DATA AND STORE APPROPRIATELY	DRT04270
C		DRT04280
	DO 300 II=1,15	DRT04290
	IF(II.EQ.15)GO TO 900	DRT04300
	READ(10IN,700)(RDIN(J),J=1,10)	DRT04310
700	FORMAT(A4,4X,9F8.2)	DRT04320
	IF(RDIN(1).EQ.RKEY(1))GO TO 50	DRT04330
	IF(RDIN(1).EQ.RKEY(2))GO TO 70	DRT04340
	IF(RDIN(1).EQ.RKEY(3))GO TO 90	DRT04350
	IF(RDIN(1).EQ.RKEY(4))GO TO 110	DRT04360
	IF(RDIN(1).EQ.RKEY(5))GO TO 130	DRT04370
	IF(RDIN(1).EQ.RKEY(6))GO TO 150	DRT04380
	IF(RDIN(1).EQ.RKEY(7))GO TO 170	DRT04390
	IF(RDIN(1).EQ.RKEY(8))GO TO 190	DRT04400
	IF(RDIN(1).EQ.RKEY(9))GO TO 210	DRT04410
	IF(RDIN(1).EQ.RKEY(10))GO TO 230	DRT04420
	IF(RDIN(1).EQ.RKEY(11))GO TO 310	DRT04430
	IF(RDIN(1).EQ.RKEY(12))GO TO 999	DRT04440
	WRITE(10OUT,804)	DRT04450
804	FORMAT(33X,52H***DIRTRAN-2 ERROR, INPUT DOES NOT CONFORM TO PROPER	DRT04460
	14H CONVENTION***)	DRT04470
	WRITE(10OUT,806)(RDIN(J),J=1,9)	DRT04480
806	FORMAT(26X,A4,4X,9F8.2)	DRT04490
	GO TO 999	DRT04500
C	STORE AND PRINT OUT ATMOSPHERIC CONDITIONS	DRT04510
C		DRT04520
	50 CONTINUE	DRT04530
	M1=.TRUE.	DRT04540
	NIO=1	DRT04550
	IF(ICLMAT.EQ.1)GO TO 55	DRT04560
	NATMOS=IFIX(RDIN(2))	DRT04570
	ZTMP(1)=RDIN(3)	DRT04580
	TMPMES(1)=RDIN(4)	DRT04590
	ZWND(1)=RDIN(5)	DRT04600
	WNDMES(1)=RDIN(6)	DRT04610
	THWND=RDIN(7)	DRT04620
	GO TO 60	DRT04630
C	IPSCAT PASQUILL CATEGORY	DRT04640
C	WINDVEL WIND VELOCITY IN M/S MEASURED AT 2 M. ABOVE GROUND	DRT04650
C	WINDIR WIND DIRECTION IN DEGREES CLOCKWISE FROM TRUE NORTH	DRT04660
C	TEMP TEMPERATURE IN DEGREES C MEASURED AT 2 M. ABOVE GROUND	DRT04670
	55 NATMOS=IPASCT	DRT04680
	ZTMP(1)=2.	DRT04690
	ZWND(1)=2.	DRT04700
	WNDMES(1)=WINDVEL	DRT04710
	TMPMES(1)=TEMP+273.0	DRT04720
	THWND=270.0-WINDIR	DRT04730
60	CONTINUE	DRT04740
	WRITE(10OUT,808)	DRT04750
		DRT04760
		DRT04770
		DRT04780
		DRT04790

808	FORMAT(1X)		DRT04800
	WRITE(IOOUT,810)PAS(NATMOS)		DRT04810
810	FORMAT(50X,28HPASQUILL CATEGORY	,A4)	DRT04820
	WRITE(IOOUT,812)(ZTMP(I),TMPMES(I),ZWND(I),WNDMES(I),I=1,NIO)		DRT04830
812	FORMAT(36X,4H HT ,F8.2,7H TEMP ,F8.2,7H HT,F8.2,7H WIND ,		DRT04840
	1 F8.2)		DRT04850
	WRITE(IOOUT,814)THWND		DRT04860
814	FORMAT(51X,22H WIND DIRECTION	,F8.2)	DRT04870
	GO TO 300		DRT04880
70	CONTINUE		DRT04890
	M2=.TRUE.		DRT04900
	ID=IFIX(RDIN(2))		DRT04910
	IF(ID.NE.0)DHDT=.TRUE.		DRT04920
	PHI=RDIN(3)		DRT04930
	WRITE(IOOUT,819)PHI		DRT04940
	IF(DHDT)WRITE(IOOUT,816)		DRT04950
816	FORMAT(47X,37H THE INVERSION LAYER HEIGHT IS GROWING)		DRT04960
	IF(.NOT.DHDT)WRITE(IOOUT,818)		DRT04970
818	FORMAT(47X,38H THE INVERSION LAYER HEIGHT IS CONSTANT)		DRT04980
819	FORMAT(7,52X,20H LATITUDE	,F8.2)	DRT04990
	GO TO 300		DRT05000
C	STORE AND WRITE SOIL CHARACTERISTICS		DRT05010
C			DRT05020
C			DRT05030
	90 CONTINUE		DRT05040
	SL=.TRUE.		DRT05050
	NSOIL=IFIX(RDIN(2))		DRT05060
	DSOD=RDIN(3)		DRT05070
	SILT=RDIN(4)		DRT05080
	IF(NSOIL.EQ.1)WRITE(IOOUT,821)		DRT05090
820	FORMAT(56X,15HSILT CONTENT	,F5.2)	DRT05100
821	FORMAT(7,63X,6HSOIL-1)		DRT05110
	IF(NSOIL.EQ.2)WRITE(IOOUT,822)		DRT05120
822	FORMAT(7,63X,6HSOIL-2)		DRT05130
	IF(NSOIL.EQ.3)WRITE(IOOUT,710)		DRT05140
710	FORMAT(7,23X,8H SOIL-3)		DRT05150
	IF(SILT.GT.1.E-06)WRITE(IOOUT,820)SILT		DRT05160
	WRITE(IOOUT,823)DSOD		DRT05170
823	FORMAT(53X,21H DEPTH OF SOD	,F5.2)	DRT05180
	IF(NSOIL.LT.1.OR.NSOIL.GT.2)NSOIL=2		DRT05190
	GO TO 300		DRT05200
C	STORE AND WRITE EXPLOSIVE CHARGE CHARACTERISTICS		DRT05210
C			DRT05220
C			DRT05230
	110 CONTINUE		DRT05240
	CH=.TRUE.		DRT05250
	NCHRG=IFIX(RDIN(2))		DRT05260
	IF(NCHRG.LT.1.OR.NCHRG.GT.5)NCHRG=1		DRT05270
	CHWT=RDIN(3)		DRT05280
	DETDEP=RDIN(4)		DRT05290
	IF(NCHRG.EQ.1)WRITE(IOOUT,824)		DRT05300
824	FORMAT(7,35X,47HSURFACE - LIVE FIRE OR 30 DEGREE TILTED STATIC,,		DRT05310
	1 14H TIP ON GROUND)		DRT05320
	IF(NCHRG.EQ.2)WRITE(IOOUT,825)		DRT05330
825	FORMAT(7,55X,22HBARE CHARGE ON SURFACE)		DRT05340
	IF(NCHRG.EQ.3)WRITE(IOOUT,826)		DRT05350
826	FORMAT(7,46X,39H30 DEGREE TILTED TIP AT 0.3 METER DEPTH)		DRT05360
	IF(NCHRG.EQ.4)WRITE(IOOUT,827)		DRT05370
827	FORMAT(7,46X,39H30 DEGREE TILTED TIP AT 0.6 METER DEPTH)		DRT05380
	IF(NCHRG.EQ.5)WRITE(IOOUT,828)		DRT05390
828	FORMAT(7,50X,32HHORIZONTAL PROJECTILE ON SURFACE)		DRT05400
	WRITE(IOOUT,829)CHWT		DRT05410
829	FORMAT(45X,30HWEIGHT OF CHARGE	,F8.2,4H KG.)	DRT05420
	WRITE(IOOUT,830)DETDEP		DRT05430
830	FORMAT(47X,30HDETONATION DEPTH	,F8.2)	DRT05440
	GO TO 300		DRT05450
C	STORE AND WRITE OUT INFORMATION ABOUT THE DETONATION LOCATIONS		DRT05460
C			DRT05470
C			DRT05480
	130 CONTINUE		DRT05490

EX=.TRUE.	DRT05500
NARY=IFIX(RDIN(2))	DRT05510
NCHS(1)=IFIX(RDIN(3))	DRT05520
NCHS(2)=IFIX(RDIN(4))	DRT05530
SRCBAS(1)=RDIN(5)	DRT05540
SRCBAS(2)=RDIN(6)	DRT05550
SIDE1(1)=RDIN(7)	DRT05560
SIDE1(2)=RDIN(8)	DRT05570
SIDE2(1)=RDIN(9)	DRT05580
SIDE2(2)=RDIN(10)	DRT05590
IF(NARY.EQ.2)GO TO 133	DRT05600
IF(NARY.EQ.3)GO TO 136	DRT05610
C	DRT05620
CC CHARGE DISTRIBUTION TYPE 1	DRT05630
C	DRT05640
WRITE(IOOUT,831)	DRT05650
831 FORMAT(/,31X,42HSIMULTANEOUS BURST , UNIFORMLY DISTRIBUTED,	DRT05660
+27H CHARGES IN A PARALLELOGRAM)	DRT05670
NCH=NCHS(1)*NCHS(2)	DRT05680
WRITE(IOOUT,832)NCH,(SRCBAS(1),I=1,2)	DRT05690
832 FORMAT(27X,28HTOTAL NUMBER OF CHARGES IS ,1X,I3,1X,	DRT05700
+27H WITH REFERENCE CHARGE AT (,F8.2,1H,,F8.2,1H))	DRT05710
WRITE(IOOUT,834)NCHS(1),(SIDE1(I),I=1,2)	DRT05720
834 FORMAT(32X,I3,1X,45HCHARGES WITH DIRECTION AND SPACING GIVEN BY (,	DRT05730
+F8.2,1H,,F8.2,1H))	DRT05740
WRITE(IOOUT,834)NCHS(2),(SIDE2(I),I=1,2)	DRT05750
GO TO 300	DRT05760
C	DRT05770
CC CHARGE DISTRIBUTION TYPE 2	DRT05780
C	DRT05790
133 NCH=NCHS(1)	DRT05800
DO 134 J=1,NCH	DRT05810
READ(IOIN,701)(COOR(K,J),K=1,2)	DRT05820
701 FORMAT(8X,2F8.2)	DRT05830
134 CONTINUE	DRT05840
WRITE(IOOUT,836)	DRT05850
836 FORMAT(/,42X,48HSIMULTANEOUS BURST, RANDOMLY DISTRIBUTED CHARGES)	DRT05860
WRITE(IOOUT,838)NCHS(1)	DRT05870
838 FORMAT(51X,26HTOTAL NUMBER OF CHARGES IS ,1X,I3)	DRT05880
WRITE(IOOUT,840)	DRT05890
840 FORMAT(55X,22HDETONATION COORDINATES)	DRT05900
DO 135 J=1,NCH	DRT05910
WRITE(IOOUT,842)(COOR(I,J),I=1,2)	DRT05920
842 FORMAT(53X,2(3X,F8.2))	DRT05930
135 CONTINUE	DRT05940
GO TO 300	DRT05950
C	DRT05960
CC CHARGE DISTRIBUTION TYPE 3	DRT05970
C	DRT05980
136 NCH=NCHS(1)	DRT05990
DO 137 J=1,NCH	DRT06000
READ(IOIN,702)(COOR(K,J),K=1,2),TSTAG(J)	DRT06010
702 FORMAT(8X,3F8.2)	DRT06020
137 CONTINUE	DRT06030
WRITE(IOOUT,844)	DRT06040
844 FORMAT(/,30X,38HSEQUENTIAL IN TIME AND RANDOM IN SPACE,	DRT06050
+24H DISTRIBUTION OF CHARGES)	DRT06060
WRITE(IOOUT,838)NCH	DRT06070
WRITE(IOOUT,846)	DRT06080
846 FORMAT(45X,25H DETONATION COORDINATES ,7X,10HBLAST TIME)	DRT06090
DO 138 J=1,NCH	DRT06100
WRITE(IOOUT,848)(COOR(I,J),I=1,2),TSTAG(J)	DRT06110
848 FORMAT(46X,F8.2,3X,F8.2,12X,F8.2)	DRT06120
138 CONTINUE	DRT06130
GO TO 300	DRT06140
C	DRT06150
CC STORE AND PRINT OUT INFORMATION ABOUT VEHICLE	DRT06160
C	DRT06170
150 CONTINUE	DRT06180
VH=.TRUE.	DRT06190

	V0(1)=RDIN(2)	DRT06200
	V0(2)=RDIN(3)	DRT06210
	VEHDIR=RDIN(4)	DRT06220
	VEHSPD=RDIN(5)	DRT06230
	VEHWID=RDIN(6)	DRT06240
	VEHWHT=RDIN(7)	DRT06250
	VEHTYP=IFIX(RDIN(8))	DRT06260
	WRITE(100UT,850)V0(1),V0(2)	DRT06270
850	FORMAT(//,44X,26HINITIAL VEHICLE POSITION (,F8.2,1H,,F8.2,1H))	DRT06280
	WRITE(100UT,852)VEHDIR	DRT06290
852	FORMAT(44X,19HVEHICLE DIRECTION ,F8.2,17H (CCW FROM EAST))	DRT06300
	WRITE(100UT,854)VEHSPD	DRT06310
854	FORMAT(50X,20HVEHICLE SPEED ,F8.2,4H M/S)	DRT06320
	WRITE(100UT,856)VEHWID	DRT06330
856	FORMAT(52X,20HVEHICLE WIDTH ,F8.2)	DRT06340
	WRITE(100UT,858)VEHWHT	DRT06350
858	FORMAT(52X,20HVEHICLE WEIGHT ,F8.2)	DRT06360
	IF(VEHTYP.EQ.0)WRITE(100UT,891)	DRT06370
891	FORMAT(58X,15HWHEELED VEHICLE)	DRT06380
	IF(VEHTYP.EQ.1)WRITE(100UT,892)	DRT06390
892	FORMAT(58X,15HTRACKED VEHICLE)	DRT06400
	GO TO 300	DRT06410
CCC	STORE TRANSMITTER COORDINATES AND TRANSMISSION THRESHOLD	DRT06420
170	CONTINUE	DRT06430
	TC=.TRUE.	DRT06440
	NEWCOR=.TRUE.	DRT06450
	TRNCOR(1)=RDIN(2)	DRT06460
	TRNCOR(2)=RDIN(3)	DRT06470
	TRNCOR(3)=RDIN(4)	DRT06480
	TRNMIN=RDIN(5)	DRT06490
	IF(TRNMIN.LT.1.E-05)TRNMIN=1.E-05	DRT06500
	THRESH=-ALOG(TRNMIN)	DRT06510
	GO TO 300	DRT06520
CCC	STORE RECEIVER COORDINATES	DRT06530
190	CONTINUE	DRT06540
	RC=.TRUE.	DRT06550
	RECCOR(1)=RDIN(2)	DRT06560
	RECCOR(2)=RDIN(3)	DRT06570
	RECCOR(3)=RDIN(4)	DRT06580
	GO TO 300	DRT06590
CCC	STORE OBSERVER COORDINATES	DRT06600
210	CONTINUE	DRT06610
	OC=.TRUE.	DRT06620
	OBSCOR(1)=RDIN(2)	DRT06630
	OBSCOR(2)=RDIN(3)	DRT06640
	SPCHT=RDIN(4)	DRT06650
	GO TO 300	DRT06660
CCC	STORE TIME INTERVAL FOR CALCULATIONS	DRT06670
230	CONTINUE	DRT06680
	TM=.TRUE.	DRT06690
	TSTART=RDIN(2)	DRT06700
	TEND=RDIN(3)	DRT06710
	TINC=RDIN(4)	DRT06720
	IF (TINC.LE.0.0) TINC=1.	DRT06730
	IF(TEND.LT.TSTART)GO TO 903	DRT06740
	LIM=IFIX((TEND-TSTART)/TINC)+1	DRT06750
300	CONTINUE	DRT06760
310	CONTINUE	DRT06770
	IF(IGEOSW.NE.1) GO TO 333	DRT06780
	TRNCOR(1)=PTS(1)*1000.	DRT06790
	TRNCOR(2)=PTS(2)*1000.	DRT06800
	TRNCOR(3)=PTS(3)*1000.	DRT06810
		DRT06820
		DRT06830
		DRT06840
		DRT06850
		DRT06860
		DRT06870
		DRT06880

```

RECCOR(1)=PTS(4)*1000.
RECCOR(2)=PTS(5)*1000.
RECCOR(3)=PTS(6)*1000.
OBSCOR(1)=PTS(10)*1000.
OBSCOR(2)=PTS(11)*1000.
SPCHT=PTS(12)*1000.
333 CONTINUE
    IOPT=IFIX(RDIN(2))
    IFILE=IFIX(RDIN(3))
    IF(IOPT.NE.3) GO TO 305
    WRITE(IOOUT,301)
    WRITE(IOOUT,302)
    WRITE(IOOUT,303) IFILE
    WRITE(IOOUT,304)
    WRITE(IOOUT,301)
301 FORMAT(1H0,130(1H*),/)
302 FORMAT(1H0,58X,15HDIRTRAN WARNING,/)
303 FORMAT(1H0,38X,13HLOGICAL UNIT ,I2,27H IS ASSIGNED TO A TEMPORARY,
+ 13H STORAGE FILE)
304 FORMAT(1H0,28X,46HCARE MUST BE TAKEN TO INSURE THAT THIS UNIT IS,
+ 27H NOT IN USE BY ANOTHER FILE,/)
305 CONTINUE
    IF((IOPT.EQ.1.OR.IOPT.EQ.2).AND.(EX.AND.(.NOT.CH)))GO TO 909
    IF((IOPT.EQ.1.OR.IOPT.EQ.2).AND.(CH.AND.(.NOT.EX)))GO TO 909
    IF((IOPT.EQ.3).AND.(OC)OC=.FALSE.)
    IF((IOPT.EQ.4).AND.(OC)OC=.FALSE.)
    IF((IOPT.EQ.5).AND.(OC)OC=.FALSE.)
    IF((TC.AND.(.NOT.RC)).OR.(RC.AND.(.NOT.TC)))GO TO 913
    IF(M1.OR.M2)NEWATM=.TRUE.
    IF(IOPT.EQ.3.AND.CH)NEWSRC=.TRUE.
    IF(EX.AND.CH)NEWSRC=.TRUE.
    IF(TC.AND.RC)LOSTRN=.TRUE.
    IF(OC)EDGE=.TRUE.
    IF(VH)NEWVEH=.TRUE.
    IF(IOPT.EQ.3)LIM=1
    DO 400 J=1,LIM
    TIME=TSTART+TINC*FLOAT(J-1)
    NEWTIM=.TRUE.
    NERR=0
    IF(IOPT.EQ.1.OR.IOPT.EQ.2)GO TO 320
    IF(IOPT.EQ.3.OR.IOPT.EQ.4)GO TO 325
    IF((IOPT.EQ.1).AND.(NARY.NE.1))GO TO 915
    IF((IOPT.EQ.2).AND.(NARY.NE.2))GO TO 915
    IF((IOPT.EQ.4).AND.(NARY.NE.3))GO TO 915
C
C
C
C
    COMPUTE FOR VEHICLE SOURCE
    CHECK TO SEE IF WE HAVE THE MINIMUM INPUT REQUIREMENTS
    IF(DSOD.GT.0.0)GO TO 315
    IF(.NOT.(M1.AND.M2.AND.SL.AND.VH.AND.TC.AND.RC.AND.TM))GO TO 911
    CALL VEHL(NATMOS,ZTMP,TMPMES,ZWND,WNDMES,THWND,PHI,NSOIL,
1      SILT,NWL,TRNCOR,RECCOR,TIME,DHDT,V0,VEHDIR,
2      VEHSPD,VEHWID,VEHWHT,VEHTYP,NEWATM,NEWVEH,TRNLOS,NERR)
    NEWVEH=.FALSE.
    NEWATM=.FALSE.
    GO TO 330
315 TRNLOS=1.0
    GO TO 335
325 CONTINUE
    IF(IOPT.EQ.4.AND.(.NOT.(EX.AND.TC.AND.RC.AND.TM)))GO TO 911
    IF(IOPT.EQ.3.AND.(.NOT.(M1.AND.M2.AND.SL.AND.CH)))GO TO 911
    CALL COMPC(NEWATM,NATMOS,ZTMP,TMPMES,ZWND,WNDMES,THWND,
1      PHI,NEWSRC,CHWT,NCHRG,NCHS,DETDEP,NSOIL,DSOD,NWL,
2      TRNCOR,RECCOR,TIME,DHDT,TRNLOS,NERR)
    NEWSRC=.FALSE.
    NEWATM=.FALSE.
    IF(IOPT.EQ.3)GO TO 410
    GO TO 330
320 CONTINUE

```

DRT06890
DRT06900
DRT06910
DRT06920
DRT06930
DRT06940
DRT06950
DRT06970
DRT06980
DRT06990
DRT07000
DRT07010
DRT07020
DRT07030
DRT07040
DRT07050
DRT07060
DRT07070
DRT07080
DRT07090
DRT07100
DRT07110
DRT07120
DRT07130
DRT07140
DRT07150
DRT07160
DRT07170
DRT07180
DRT07190
DRT07200
DRT07210
DRT07220
DRT07230
DRT07240
DRT07250
DRT07260
DRT07270
DRT07280
DRT07290
DRT07300
DRT07310
DRT07320
DRT07330
DRT07340
DRT07350
DRT07360
DRT07370
DRT07380
DRT07390
DRT07400
DRT07410
DRT07420
DRT07430
DRT07440
DRT07450
DRT07460
DRT07470
DRT07480
DRT07490
DRT07500
DRT07510
DRT07520
DRT07530
DRT07540
DRT07550
DRT07560
DRT07570
DRT07580
DRT07590

C	CHECK TO SEE IF MINIMUM INPUTS ARE AVAILABLE	DRT07600
C	IF(.NOT.(M1.AND.M2.AND.SL.AND.CH.AND.EX.AND.(TC.AND.RC.OR.OC)).AND.	DRT07610
C	1TM)GO TO 911	DRT07620
	CALL DUSTCL(NEWATM,NATMOS,ZTMP,TMPMES,ZWWD,WNDMES,PHI,	DRT07630
1	THWWD,NEWSRC,CHWT,NCHRG,DETDEP,NSOIL,DSOD,	DRT07640
2	LOSTRN,TRNCOR,RECCOR,EDGE,OBSCOR,SPCHT,NEWTIM,	DRT07650
3	TIME,TRNLOS,CNTRD,HEIGHT,CENWTH,SPCWTH,NCPTS,CPTS,	DRT07660
4	NERR,NCHS,SRCBAS,SIDE1,SIDE2,DHDT)	DRT07670
	NEWSRC=.FALSE.	DRT07680
	NEWATM=.FALSE.	DRT07690
330	IF(NERR.EQ.0)GO TO 335	DRT07700
	WRITE(IOOUT,857)NERR	DRT07710
857	FORMAT(55X,30H ***** DIRTRAN ERROR NUMBER ,12)	DRT07720
	GO TO 400	DRT07730
335	IF(ONCE.AND.(.NOT.NEWCOR))GO TO 340	DRT07740
	NEWCOR=.FALSE.	DRT07750
	IZINV=IFIX(ZINV)	DRT07760
	WRITE(IOOUT,859)IZINV	DRT07770
859	FORMAT(//,47X,30HESTIMATED INVERSION HEIGHT ,17)	DRT07780
C	IF OBSERVER IS SPECIFIED, OUTPUT IS LABELED FOR EACH TIME.	DRT07790
C	IF ONLY TRANSMITTER AND RECEIVER ARE INPUT, OUTPUT IS TABULAR	DRT07800
C	IF(TC.AND.RC.AND.OC)GO TO 350	DRT07810
	IF(OC)GO TO 350	DRT07820
	IF(.NOT.(TC.AND.RC))GO TO 905	DRT07830
	WRITE(IOOUT,860)WAVE1	DRT07840
860	FORMAT(//,47X,18HWAVELENGTH ,F7.2,12H MICROMETERS)	DRT07850
	WRITE(IOOUT,862)(TRNCOR(I),I=1,3)	DRT07860
	WRITE(IOOUT,864)(RECCOR(I),I=1,3)	DRT07870
862	FORMAT(37X,28HTRANSMITTER COORDINATES ,3F10.2)	DRT07880
864	FORMAT(37X,28HRECEIVER COORDINATES ,3F10.2)	DRT07890
	WRITE(IOOUT,866)	DRT07900
866	FORMAT(52X,4HTIME,10X,13HTRANSMITTANCE)	DRT07910
340	CONTINUE	DRT07920
	WRITE(IOOUT,868)TIME,TRNLOS	DRT07930
868	FORMAT(52X,F8.2,10X,E10.5)	DRT07940
	ONCE=.TRUE.	DRT07950
	GO TO 400	DRT07960
350	WRITE(IOOUT,923)TIME	DRT07970
923	FORMAT(//,48X,28HTIME AFTER BLAST ,F7.2)	DRT07980
	IF(.NOT.(TC.AND.RC))GO TO 360	DRT07990
	WRITE(IOOUT,808)	DRT08000
	WRITE(IOOUT,860)WAVE1	DRT08010
	WRITE(IOOUT,862)(TRNCOR(I),I=1,3)	DRT08020
	WRITE(IOOUT,864)(RECCOR(I),I=1,3)	DRT08030
	WRITE(IOOUT,870)TRNLOS	DRT08040
870	FORMAT(42X,38HTRANSMITTANCE ALONG THE LINE OF SIGHT ,E10.3)	DRT08050
360	WRITE(IOOUT,808)	DRT08060
	WRITE(IOOUT,872)	DRT08070
872	FORMAT(57X,28HAERODYNAMIC CLOUD DIMENSIONS)	DRT08080
	WRITE(IOOUT,808)	DRT08090
	WRITE(IOOUT,874)(OBSCOR(I),I=1,2)	DRT08100
874	FORMAT(41X,28HOBSERVER COORDINATES ,2F10.2)	DRT08110
	WRITE(IOOUT,876)HEIGHT	DRT08120
876	FORMAT(39X,26HTHE HEIGHT OF THE CLOUD IS,10X,F10.2,7H METERS)	DRT08130
	WRITE(IOOUT,878)(CNTRD(I),I=1,2)	DRT08140
878	FORMAT(38X,28HTHE CENTROID COORDINATES ARE,8X,2F10.2)	DRT08150
	WRITE(IOOUT,880)CENWTH	DRT08160
880	FORMAT(38X,28HTHE WIDTH AT THE CENTROID IS,8X,F10.2, 7H METERS)	DRT08170
	WRITE(IOOUT,882)SPCHT,SPCWTH	DRT08180
882	FORMAT(39X,12HTHE WIDTH AT,F8.2,11H METERS IS ,5X,F10.2,7H METERS)	DRT08190
	WRITE(IOOUT,884)NCPTS	DRT08200
884	FORMAT(46X,13,37H CONTOUR POINTS HAVE BEEN DETERMINED)	DRT08210
	WRITE(IOOUT,886)((CPTS(I,IPT),I=1,2),IPT=1,NCPTS)	DRT08220
886	FORMAT(60X,2(F10.3,2X))	DRT08230
400	CONTINUE	DRT08240
	GO TO 10	DRT08250
		DRT08260
		DRT08270
		DRT08280
		DRT08290

410 WRITE(1000,888)	DRT08300
888 FORMAT(/,38X,48H** THE CLOUD HAS BEEN PRECOMPUTED AND STORED ON,	DRT08310
1 5H FILE)	DRT08320
GO TO 10	DRT08330
900 WRITE(1000,901)	DRT08340
901 FORMAT(/,24X,48H*** DIRTRAN ERROR - MORE THAN 15 RECORDS OF DATA,	DRT08350
1 35H HAVE BEEN INPUT WITHOUT A GO CARD.)	DRT08360
IERR=1	DRT08370
GO TO 999	DRT08380
903 WRITE(1000,904)	DRT08390
904 FORMAT(/,39X,47H*** DIRTRAN ERROR - TIMES ARE NOT IN INCREASING,	DRT08400
+ 6H ORDER)	DRT08410
IERR=1	DRT08420
GO TO 999	DRT08430
905 WRITE(1000,906)	DRT08440
906 FORMAT(/,18X,46H*** DIRTRAN ERROR -NO TRANSMITTER AND RECEIVER,	DRT08450
+ 49H AND/OR OBSERVER COORDINATES HAVE BEEN SPECIFIED.)	DRT08460
IERR=1	DRT08470
GO TO 999	DRT08480
909 WRITE(1000,910)	DRT08490
910 FORMAT(/,25X,44H*** DIRTRAN ERROR - ONLY ONE DATA RECORD FOR,	DRT08500
1 38H CHARGE INFORMATION HAS BEEN SPECIFIED)	DRT08510
IERR=1	DRT08520
GO TO 999	DRT08530
911 WRITE(1000,912)	DRT08540
912 FORMAT(/,16X,49H*** DIRTRAN ERROR - MINIMUM AMOUNT OF INFORMATION,	DRT08550
1 26H REQUIRED IS NOT AVAILABLE./,10X,14H CHECK INPUTS)	DRT08560
IERR=1	DRT08570
GO TO 999	DRT08580
913 WRITE(1000,914)	DRT08590
914 FORMAT(/,20X,49H*** DIRTRAN ERROR - BOTH TRANSMITTER AND RECEIVER,	DRT08600
1 43H LOCATIONS MUST BE SPECIFIED, CHECK INPUTS)	DRT08610
IERR=1	DRT08620
GO TO 999	DRT08630
915 WRITE(1000,916)	DRT08640
916 FORMAT(/,30X,50H IOPT AND NARY DO NOT AGREE SEE THE ABOVE COMMENTS,	DRT08650
1 21H FOR CORRECT MATCHING)	DRT08660
IERR=1	DRT08670
999 RETURN	DRT08680
END	DRT08690

SUBROUTINE AMOUNT(VOLSPH,WAKAL,SPHAL) SUBROUTINE TO DETERMINE LOADING FOR THE SPHERE AND WAKE INPUTS VOLSPH - VOLUME OF THE BUOYANTLY RISING SPHERE ALL OTHER NEEDED INFORMATION IS PASSED VIA COMMON BLOCKS OUTPUTS WAKAL - AMOUNT OF INITIAL LOADING OF SPHERE THAT HAS BEEN DEPOSITED IN THE WAKE SPHAL - AMOUNT OF INITIAL LOADING OF SPHERE THAT IS LEFT IN THE SPHERE FUNCTIONS AND SUBROUTINES NEEDED NONE ***** COMMON/NTAL/TNOT,VOLNOT,TNO,CBLEED COMMON/BUOYCL/RSPH,DELT,VZ,XCM,YCM,ZCM,XTOP,YPOT,SPHNS(3),RISTIM TSPH=TNO+DELT STUFF=CBLEED*(VOLSPH/TSPH-VOLNOT/TNOT) WAKAL=AMIN(1,STUFF) SPHAL=SPHNS(1)-WAKAL RETURN END	AMOU0250 AMOU0010 AMOU0020 AMOU0030 AMOU0040 AMOU0050 AMOU0060 AMOU0070 AMOU0080 AMOU0090 AMOU0100 AMOU0110 AMOU0120 AMOU0130 AMOU0140 AMOU0150 AMOU0160 AMOU0170 AMOU0180 AMOU0190 AMOU0200 AMOU0210 AMOU0220 AMOU0230 AMOU0240 AMOU0260 AMOU0270 AMOU0280 AMOU0290 AMOU0300 AMOU0310 AMOU0320 AMOU0330
--	--


```

SUBROUTINE ATMCAL(NATM,ZT,TMES,ZU,UMES,PHI,BETA,DHDT,ERR)
REAL M,N,K,KM
LOGICAL ERR,DHDT
DIMENSION ZT(2),TMES(2),ZU(2),UMES(2),ZL0(6)
COMMON /WINDPRM/ DXZ0,DYX0,DZ0,U0,M,N,ZINV
COMMON /EKTEMP/ Z0,ZL,T0,TC1,TC2,TC3
COMMON /EKWIND/ ALP,C,PYF,PXF,UHAT,VHAT
COMMON /STARS/ USTAR,TSTAR,ZSTAR
COMMON /IOUNIT/ IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
COMMON /CONST/ PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK
DATA ZL0/-2.5,-4.5,-13.5,10000.,55.,20./
DATA OMEGA,K /7.2722E-05,4/

```

PURPOSE

TO FIT THE BEST POWER-LAW PROFILES OF WINDSPEED AND DIFFUSIVITY CONSISTENT WITH KNOWN RELATIONS GOVERNING THE CONSTANT SHEAR STRESS LAYER TO GIVEN MEASUREMENTS AT ONE OR TWO HEIGHTS. ALSO TO CALCULATE PARAMETERS NEEDED FOR VERTICAL VARIATION IN WIND DIRECTION LAYER, FOR WIND AND TEMPERATURE PROFILES.

INPUTS

NATM INTEGER WHICH IS 0 IF WINDSPEED AND TEMPERATURE ARE AVAILABLE AT TWO HEIGHTS AND EQUAL TO THE PASQUILL CATEGORY OTHERWISE.

ZT SINGLY DIMENSIONED ARRAY CONTAINING TWO HEIGHTS (IN METERS) AT WHICH TEMPERATURES WILL BE GIVEN. MUST BE IN ASCENDING ORDER.

TMES SINGLY DIMENSIONED ARRAY CONTAINING THE TWO TEMPERATURE MEASUREMENTS IN DEGREES KELVIN AT HEIGHTS ZT.

ZU SINGLY DIMENSIONED ARRAY CONTAINING ONE OR TWO HEIGHTS (METERS) AT WHICH WIND SPEEDS WILL BE GIVEN. MUST BE IN ASCENDING ORDER.

UMES SINGLY DIMENSIONED ARRAY CONTAINING THE ONE OR TWO WIND SPEED MEASUREMENTS (M/S) AT HEIGHTS UMES.

PHI LATITUDE OF DETONATION SITE.

BETA ANGLE OF WIND VELOCITY VECTOR MEASURED COUNTER-CLOCKWISE FROM EAST.

DHDT A LOGICAL VARIABLE WHICH IS .FALSE. IF THE INVERSION LAYER HEIGHT IS RELATIVELY CONSTANT AND .TRUE. IF THE LAYER HEIGHT IS INCREASING.

OUTPUTS

ERR A LOGICAL WHICH IS TRUE IF AN ERROR IS INCURRED DURING THE CALCULATION.

DXZ0 THE RATIO OF THE DIFFUSIVITY IN THE X DIRECTION TO THE DIFFUSIVITY IN THE Z DIRECTION. RETURNED IN COMMON /WINDPRM/.

DYX0 THE RATIO OF THE DIFFUSIVITY IN THE Y DIRECTION TO THE DIFFUSIVITY IN THE X DIRECTION. RETURNED IN COMMON /WINDPRM/.

ATMC0010
ATMC0020
ATMC0030
ATMC0040
ATMC0050
ATMC0060
ATMC0070
ATMC0080
ATMC0090
ATMC0100
ATMC0110
ATMC0120
ATMC0130
ATMC0140
ATMC0150
ATMC0160
ATMC0170
ATMC0180
ATMC0190
ATMC0200
ATMC0210
ATMC0220
ATMC0230
ATMC0240
ATMC0250
ATMC0260
ATMC0270
ATMC0280
ATMC0290
ATMC0300
ATMC0310
ATMC0320
ATMC0330
ATMC0340
ATMC0350
ATMC0360
ATMC0370
ATMC0380
ATMC0390
ATMC0400
ATMC0410
ATMC0420
ATMC0430
ATMC0440
ATMC0450
ATMC0460
ATMC0470
ATMC0480
ATMC0490
ATMC0500
ATMC0510
ATMC0520
ATMC0530
ATMC0540
ATMC0550
ATMC0560
ATMC0570
ATMC0580
ATMC0590
ATMC0600
ATMC0610
ATMC0620
ATMC0630
ATMC0640
ATMC0650
ATMC0660
ATMC0670
ATMC0680
ATMC0690
ATMC0700

DZ0	THE COEFFICIENT OF Z**N IN THE VERTICAL PROFILE OF VERTICAL DIFFUSIVITY. RETURNED IN COMMON /WNDPRM/.	ATMC0710
U0	THE COEFFICIENT OF Z**M IN THE VERTICAL PROFILE OF HORIZONTAL WIND SPEED. RETURNED IN COMMON /WNDPRM/.	ATMC0720
M	THE EXPONENT OF Z IN THE HORIZONTAL WIND SPEED PROFILE. RETURNED IN COMMON /WNDPRM/.	ATMC0730
N	THE EXPONENT OF Z IN THE VERTICAL DIFFUSIVITY PROFILE. RETURNED IN COMMON /WNDPRM/.	ATMC0740
ZINV	ESTIMATED INVERSION HEIGHT. RETURNED IN /WNDPRM/.	ATMC0750
USTAR	VELOCITY PROFILE SCALE RETURNED IN COMMON /STARS/.	ATMC0760
TSTAR	TEMPERATURE PROFILE SCALE. RETURNED IN COMMON /STARS/.	ATMC0770
ZSTAR	HEIGHT AT WHICH THE VERTICAL VARIATION IN WIND DIRECTION PROFILES FOR WIND AND TEMPERATURE TAKE EFFECT.	ATMC0780

OUTPUT RETURNED IN COMMON /EKWIND/ ARE THE PARAMETERS NEEDED FOR THE WIND PROFILE ABOVE ZSTAR.

OUTPUT RETURNED IN COMMON /EKTEMP/ ARE THE PARAMETERS NEEDED FOR THE TEMPERATURE PROFILES ABOVE ZSTAR.

CALLED FROM DUSTCL

NEEDED FUNCTIONS AND SUBROUTINES

TMPCL	CALCULATES SCALED TEMPERATURE PROFILES	ATMC0790
WINDCL	CALCULATES SCALED WIND SPEED PROFILES	ATMC0800
DIFFUS	FUNCTION TO CALCULATE THE DIFFUSIVITY AT A GIVEN HEIGHT.	ATMC0810
TEMP	CALCULATES THE POTENTIAL TEMPERATURE AND GRADIENT AT A GIVEN HEIGHT.	ATMC0820

ERR=.FALSE.

DELTH IS THE DIFFERENCE IN POTENTIAL TEMPERATURE BETWEEN THE TWO HEIGHTS WHERE TEMPERATURE IS GIVEN.

Z0=0.01

T0=TMES(1)

IF(NATM.EQ.0)GO TO 100

ASSIGN ATMOSPHERIC PROFILE ACCORDING TO SPECIFIED PASQUILL CATEGORY

Z0	FRICTION HEIGHT	ATMC0830
ZL	MONIN-OBUKOV LENGTH	ATMC0840
USTAR	THE FRICTION VELOCITY	ATMC0850
TSTAR	THE SCALING TEMPERATURE	ATMC0860

ZL=ZL0(NATM)

IF(NATM.GE.5)Z0=1.E-04*ABS(ZL)

IF(NATM.LE.3)Z0=1.E-03*ABS(ZL)

NP=IFIX(SIGN(1.,ZL))

ATMC0790
ATMC0800
ATMC0810
ATMC0820
ATMC0830
ATMC0840
ATMC0850
ATMC0860
ATMC0870
ATMC0880
ATMC0890
ATMC0900
ATMC0910
ATMC0920
ATMC0930
ATMC0940
ATMC0950
ATMC0960
ATMC0970
ATMC0980
ATMC0990
ATMC1000
ATMC1010
ATMC1020
ATMC1030
ATMC1040
ATMC1050
ATMC1060
ATMC1070
ATMC1080
ATMC1090
ATMC1100
ATMC1110
ATMC1120
ATMC1130
ATMC1140
ATMC1150
ATMC1160
ATMC1170
ATMC1180
ATMC1190
ATMC1200
ATMC1210
ATMC1220
ATMC1230
ATMC1240
ATMC1250
ATMC1260
ATMC1270
ATMC1280
ATMC1290
ATMC1300
ATMC1310
ATMC1320
ATMC1330
ATMC1340
ATMC1350
ATMC1360
ATMC1370
ATMC1380
ATMC1390
ATMC1400

	USTAR=UMES(1)/WINDCAL(Z0,ZL,ZU(1))	ATMC1410
	TSTAR=TMES(1)*USTAR**2/1.568/ZL	ATMC1420
	IF(NATM-4)200,300,210	ATMC1430
100	CONTINUE	ATMC1440
C		ATMC1450
C	USE ITERATIVE PROCEDURE TO CONVERGE ON BEST ATMOSPHERIC PROFILE	ATMC1460
C	TO MATCH DATA AT TWO HEIGHTS	ATMC1470
	DELTH=TMES(2)-TMES(1)+.0098*(ZT(2)-ZT(1))	ATMC1480
	NP=SIGN(1.,DELTH)	ATMC1490
	DELU=UMES(2)-UMES(1)	ATMC1500
	ZULOG=ALOG(ZU(2)/ZU(1))	ATMC1510
	ZTLOG=ALOG(ZT(2)/ZT(1))	ATMC1520
	USTAR=(UMES(2)-UMES(1))/ZULOG	ATMC1530
	TSTAR=DELTH/ZTLOG	ATMC1540
	ZL=.638*TMES(1)*USTAR**2/TSTAR	ATMC1550
	IF(ABS(ZL).GE.1000.)GO TO 300	ATMC1560
	DO 110 ITER=1,100	ATMC1570
	USTAR=DELU/(WINDCAL(Z0,ZL,ZU(2))-WINDCAL(Z0,ZL,ZU(1)))	ATMC1580
	TSTAR=DELTH/(TMPCAL(Z0,ZL,ZT(2))-TMPCAL(Z0,ZL,ZT(1)))	ATMC1590
	ZLP=ZL	ATMC1600
	ZL=.638*TMES(1)*USTAR**2/TSTAR	ATMC1610
	IF(ABS(ZL-ZLP)/ZLP).LT..01)GO TO 120	ATMC1620
110	CONTINUE	ATMC1630
	ERR=.TRUE.	ATMC1640
	GO TO 999	ATMC1650
120	CONTINUE	ATMC1660
	IF(ZL.GT.0.)Z0=1.E-04*ABS(ZL)	ATMC1670
	IF(ZL.LE.0.)Z0=1.E-03*ABS(ZL)	ATMC1680
	IF(NP)200,300,210	ATMC1690
200	CONTINUE	ATMC1700
C		ATMC1710
C	UNSTABLE ATMOSPHERE	ATMC1720
	DXZ0=2.6	ATMC1730
	M=.079943	ATMC1740
	N=4./3.	ATMC1750
	DZ0=.7609*USTAR*ABS(ZL)**(1.-N)	ATMC1760
	U0=USTAR*14.2478/ABS(ZL)**M	ATMC1770
	GO TO 430	ATMC1780
210	CONTINUE	ATMC1790
C		ATMC1800
C	STABLE ATMOSPHERE	ATMC1810
	DXZ0=3.3	ATMC1820
	N=.45644	ATMC1830
	M=.28414	ATMC1840
	DZ0=.059517*USTAR*ABS(ZL)**(1.-N)	ATMC1850
	U0=USTAR*36.6642/ABS(ZL)**M	ATMC1860
	GO TO 430	ATMC1870
300	CONTINUE	ATMC1880
C		ATMC1890
C	NEUTRAL ATMOSPHERE	ATMC1900
	DZ0=.4*USTAR	ATMC1910
	DXZ0=2.8	ATMC1920
	NP=0	ATMC1930
	N=1.	ATMC1940
	M=1./7.	ATMC1950
	U0=45.92*USTAR/ABS(ZL)**M	ATMC1960
430	CONTINUE	ATMC1970
C		ATMC1980
C	COMMON CALCULATION TO UNSTABLE, NEUTRAL, AND STABLE ATMOSPHERES	ATMC1990
	DYX0=1.	ATMC2000
	IF(NATM.EQ.0)U0=(U0+UMES(2)/ZU(2)**M)/2.	ATMC2010
C		ATMC2020
C	ESTIMATE THE INVERSION HEIGHT AND COMPUTE THE NECESSARY	ATMC2030
	PARAMETERS FOR THE WIND AND TEMPERATURE PROFILES BETWEEN	ATMC2040
	ZSTAR AND ZINV WHEN DHDOT IS .FALSE..	ATMC2050
C		ATMC2060
C		ATMC2070
C		ATMC2080
		ATMC2090
		ATMC2100

C

```

APHI=PHI*PIRAD
FREQ=2.*OMEGA*SIN(APHI)
HC=K*USTAR/FREQ
IF(ZL.GT.0.0.AND.ZL.LT.1.E3)GO TO 500
ZINV=HC
GO TO 501
500 ZINV=.26*HC
501 ZSTAR=.13*ZINV
KM=DIFFUS(Z0,ZL,ZSTAR)
SP=USTAR*WINDCAL(Z0,ZL,ZSTAR)
ALP=SQRT(FREQ/(2.*KM))
IF(DHOT)GO TO 813
ARG=ALP*ZSTAR
ARG1=BETA+ALP*ZINV
ARG2=ALP*(ZINV-ZSTAR)
C=SP*EXP(ARG)*SIN(ARG1)/SIN(ARG2)
PYF=C*EXP(-ARG)*COS(ARG)-SP*COS(BETA)
PXF=C*EXP(-ARG)*SIN(ARG)+SP*SIN(BETA)
UE=C*EXP(-ARG)*COS(ARG)-PYF
VE=-C*EXP(-ARG)*SIN(ARG)+PXF
UHAT=UE/SQRT(UE*UE+VE*VE)
VHAT=VE/SQRT(UE*UE+VE*VE)
CALL TEMP(ZSTAR,TA,DTADZ)
DTADH=0.0
TC3=(DTADH-DTADZ)/(2.*(ZINV-ZSTAR))
TC2=DTADZ-2.*TC3*ZSTAR
TC1=TA-TC2*ZSTAR-TC3*ZSTAR**2
GO TO 999
813 ZSTAR=1.E4
999 RETURN
END

```

```

ATMC2110
ATMC2120
ATMC2130
ATMC2140
ATMC2150
ATMC2160
ATMC2170
ATMC2180
ATMC2190
ATMC2200
ATMC2210
ATMC2220
ATMC2230
ATMC2240
ATMC2250
ATMC2260
ATMC2270
ATMC2280
ATMC2290
ATMC2300
ATMC2310
ATMC2320
ATMC2330
ATMC2340
ATMC2350
ATMC2360
ATMC2370
ATMC2380
ATMC2390
ATMC2400
ATMC2410
ATMC2420

```

```

SUBROUTINE AVRG(ZX,TIME,QTOT,XBAVG,SIG2X,SIG2Y)
ROUTINE FOR FINDING AVERAGES OF THE MOMENTS OF THE DISCS
INPUTS
  ZX - HEIGHT AT WHICH THE AVERAGES ARE DESIRED
  TIME - PRESENT TIME
  QTOT - SUM OF T00*QDSCS
  XBAVG- AVERAGE OF THE FIRST ORDER MOMENTS (ALONGWIND DISPLACEMENT
         IN THE WIND DIRECTION)
  SIG2X- AVERAGE OF THE SQUARE OF ONE OF THE SECOND ORDER MOMENTS
         (ALONGWIND SPREAD)
  SIG2Y- AVERAGE OF THE SQUARE OF ANOTHER SECOND ORDER MOMENT
         (CROSSWIND SPREAD)
*****
COMMON/DISCS/NDSCS,TDSC(20),XDSC(20),ZDSC(20),R2DSC(20),QDSC(20,3)
COMMON/PRTIME/R0,VGRAY(3),NPRTS
QTOT=0.0
QSIG2X=0.0
QSIG2Y=0.0
QXBAR=0.0
Z=ZX
DO 10 I=1,NDSCS
  H=ZDSC(I)
  ROH2=R2DSC(I)
  TOF=TIME-TDSC(I)
  CALL MOMENT(VGRAY(1),Z,H,TOF,Q,XBAR,SIGW2,SIGP2)
  QT=QDSC(I,1)*Q
  QTOT=QTOT+QT
  QSIG2X=QSIG2X+(SIGW2+ROH2/2.)*QT
  QSIG2Y=QSIG2Y+(SIGP2+ROH2/2.)*QT
  QXBAR=QXBAR+(XBAR+XDSC(I))*QT
10 CONTINUE
XBAVG=QXBAR/QTOT
SIG2X=QSIG2X/QTOT
SIG2Y=QSIG2Y/QTOT
999 RETURN
END

```

```

AVRG0230
AVRG0010
AVRG0020
AVRG0030
AVRG0040
AVRG0050
AVRG0060
AVRG0070
AVRG0080
AVRG0090
AVRG0100
AVRG0110
AVRG0120
AVRG0130
AVRG0140
AVRG0150
AVRG0160
AVRG0170
AVRG0180
AVRG0190
AVRG0200
AVRG0210
AVRG0220
AVRG0240
AVRG0250
AVRG0260
AVRG0270
AVRG0280
AVRG0290
AVRG0300
AVRG0310
AVRG0320
AVRG0330
AVRG0340
AVRG0350
AVRG0360
AVRG0370
AVRG0380
AVRG0390
AVRG0400
AVRG0410
AVRG0420
AVRG0430
AVRG0440
AVRG0450
AVRG0460

```

```

SUBROUTINE CLIMB(FUNCT,GFUN,P1,FP1,NSERCH,NOCONT)
  THIS MODULE IS A SUBROUTINE THAT FINDS A POINT ON A CONTOUR
  BY FINDING THE GRADIENT VECTOR AT THAT POINT AND MARCHING ALONG
  IT UNTIL IT FINDS ITSELF IN A REGION GREATER THAN THE CONTOUR LEVEL.
  AT WHICH POINT IT MARCHES HORIZONTALLY, HALVING THE STEP SIZE
  UNTIL THE CONTOUR IS REACHED WITHIN SPECIFIED RESOLUTION.
  IN ADDITION IT WILL DETERMINE IF A CONTOUR EXISTS.

  ARGUMENTS PASSED.

  INPUT
    FUNCT-THE FUNCTION(X,Y) ALSO GIVEN IN EXTERNAL.
    P1-THE STARTING POINT.

  OUTPUT
    P1 - THE POINT ON THE CONTOUR OR THE POINT AT WHICH
        THE FUNCTION REACHES A MAXIMUM BELOW THE CONTOUR
        LEVEL
    FP1 - THE VALUE OF THE FUNCTION AT P
    NOCONT-THE ERROR FLAG.
        F-NO PROBLEM
        T-NO CONTOUR FOUND.
    ERR-ERROR FLAG RETURNED BY 'NTRSC'
        F-NO ERROR
        T-ITERATION DIVERGED OR MAXIMUM SEARCH AREA EXCEEDED

  IN ADDITION,IN COMMON ARE...
    YMIN-THE LOWER LIMIT ON Y.
    DELTA-THE STEP SIZE,MODIFIED IN THIS SUBROUTINE.
    CON-THE CONTOUR LEVEL.
    RES-THE RESOLUTION LENGTH

  OTHER VARIABLES INCLUDE
    GRAD-THE GRADIENT VECTOR
    P0-THE CURRENT POINT ON THE GRADIENT.
    P1-THE POINT ON THE GRADIENT BEING TESTED
        TO SEE ABOUT CONTOUR EXISTENCE.
    FP0,FP1-THE FUNCTION VALUES OF P0 AND P1.

  CALLED SUBROUTINES
    GRAD2-FINDS THE GRADIENT VECTOR OF A FUNCTION AT
        A POINT AND THE SLOPE THERE.
    UNIT-CALCULATES THE NORM AND MAGNITUDE OF A 2 VECTOR.
    VSUM-VECTOR SUM OF THE FORM C=A+SB WHERE S IS SCALAR
        MULTIPLIER OF B.

  EXTERNAL FUNCT
  LOGICAL NOCONT
  DIMENSION GRAD(2),P0(2),P1(2)
  COMMON/LINE/BASE(2),DIR(2),DFDS/SPECS/RES,DELTA,THETAN,CON
  COMMON/LIMIT/YMIN,FMIN
  NOCONT=.FALSE.
  ONEM=-1.0
  IF (NSERCH.EQ.0)GO TO 7
  DELTA=SIGN(DELTA, FLOAT(NSERCH))
  FP1=FUNCT(P1(1),P1(2))
  IF(FP1.LT.CON)GO TO 25
  GO TO 22
3  CONTINUE
  P0(1)=P1(1)
  P0(2)=P1(2)
  FP0=FP1
C ** FINDING THE UNIT GRADIENT AND THE NEXT POINT ALONG IT.
  4 CALL GRAD2(P0,FUNCT,RES,GRAD,DFDS)

```

```

CL100520
CL100010
CL100020
CL100030
CL100040
CL100050
CL100060
CL100070
CL100080
CL100090
CL100100
CL100110
CL100120
CL100130
CL100140
CL100150
CL100160
CL100170
CL100180
CL100190
CL100200
CL100210
CL100220
CL100230
CL100240
CL100250
CL100260
CL100270
CL100280
CL100290
CL100300
CL100310
CL100320
CL100330
CL100340
CL100350
CL100360
CL100370
CL100380
CL100390
CL100400
CL100410
CL100420
CL100430
CL100440
CL100450
CL100460
CL100470
CL100480
CL100490
CL100500
CL100510
CL100530
CL100540
CL100550
CL100560
CL100570
CL100580
CL100590
CL100600
CL100610
CL100620
CL100630
CL100640
CL100650
CL100660
CL100670
CL100680
CL100690
CL100700

```

```

      5 CALL VSUM(P0,GRAD,DELTA,P1)
C ** IS THE POINT HEADING BELOW YMIN **
      IF(P1(2).GE.YMIN)GO TO 7
      P1(2)=YMIN
      CALL VSUM(P1,P0,ONEM,GRAD)
      CALL UNIT(GRAD,GRAD,DELTA)
      IF(ABS(DELTA).LT.RES)GO TO 25
      7 FP1=FUNCT(P1(1),P1(2))
C ** HAS THE CONTOUR BEEN CROSSED **
      8 IF(FP1.GE.CON)GO TO 22
      IF(FP1.GT.FP0)GO TO 3
      DELTA=DELTA/2.
      IF(ABS(DELTA).LT.RES)GO TO 25
      GO TO 5
      25 NOCONT=.TRUE.
      GO TO 99
      22 CONTINUE
C BEGIN HORIZONTAL SEARCH
      P0(2)=P1(2)
      31 P0(1)=P1(1)
      FP0=FP1
      40 P1(1)=P0(1)+DELTA
      FP1=FUNCT(P1(1),P1(2))
      IF(ABS(DELTA).LT.RES/2.)GO TO 99
      IF(FP1.GE.CON)GO TO 31
      DELTA=DELTA/2.
      GO TO 40
      99 CONTINUE
      RETURN
      END

```

```

CL100710
CL100720
CL100730
CL100740
CL100750
CL100760
CL100770
CL100780
CL100790
CL100800
CL100810
CL100820
CL100830
CL100840
CL100850
CL100860
CL100870
CL100880
CL100890
CL100900
CL100910
CL100920
CL100930
CL100940
CL100950
CL100960
CL100970
CL100980
CL100990
CL101000

```

SUBROUTINE CLDIM(CNTRD,HEIGHT,CENWTH,SPCHT,SPCWTH,NCPTS,CPTS5,
1 ERR)

PURPOSE

CLDIM CALCULATES FIVE CONTOUR POINTS AND CLOUD DIMENSIONS AS
SEEN FROM THE SPECIFIED OBSERVER POSITION. CLDIM REQUIRES CLOUD
PARAMETERS FROM THE BUOYANT RISE STAGE OF CLOUD DEVELOPMENT WHICH
ARE SUPPLIED IN COMMON STORAGE /BUOYCL/ AND /PRTINF/ AS WELL AS
VIEWING GEOMETRY WHICH IS SUPPLIED IN COMMON /GEOM/. SPCHT IS
REQUIRED INPUT IN THE ARGUMENTS. ALL OUTPUTS ARE ARGUMENTS.

INPUT

SPCHT THE SPECIFIED HEIGHT AT WHICH THE WIDTH OF THE CLOUD
IS DESIRED. (METERS)

OUTPUT

CNTRD A SINGLY DIMENSIONED ARRAY OF LENGTH 2 WHICH CONTAINS
THE HORIZONTAL AND VERTICAL COORDINATES, RESP., OF THE
CLOUD CENTROID. (METERS)
HEIGHT THE HEIGHT OF THE CLOUD IN METERS
CENWTH THE WIDTH OF THE CLOUD AT THE CENTROID HEIGHT IN METERS
SPCWTH THE WIDTH OF THE CLOUD AT THE SPECIFIED HEIGHT (METERS)
NCPTS THE NUMBER OF CONTOUR POINTS (=6)
CPTS A DOUBLY DIMENSIONED ARRAY OF SIZE (2,N),N.GE.5, WHICH
CONTAINS THE HORIZONTAL AND VERTICAL COORDINATES OF
THE FIVE CONTOUR POINTS. (METERS)

REQUIRED SUBROUTINES

CLIMB DETERMINES IF THE CONTOUR EXISTS, IF SO FINDS A POINT
ON THE CONTOUR.

CALLED BY DUSTCL

DIMENSION CNTRD(2),CPTS5(2,6),TOP(2)
LOGICAL HORIZ,NOCONT,SWITCH,CHANGE,ERR
REAL KZ,KX
COMMON /BUOYCL/RSPH,DELT,VZ,XCM,YCM,ZCM,XTOP,YTOP,SPHNS(3),TIM
COMMON /PRTINF/R0,VGRAY(3),NPRTS
COMMON /GEOM/COSTH2,SINTH,SINTH2,VISEXT,RTPI,SCRN(2)
COMMON /MODE/ HORIZ
COMMON /CLOCK/ T,TWIND
COMMON /ARRAY/OVRLAP,AREA,PERIM,PRJARY,CENDIF
COMMON /WINDFRM/DXZ0,DYX0,DZ0,U0,UM,DN,ZINV
COMMON /TRAN/VTR,KZ,KX,TTR,XTR,ZTR,QPUFF(3),SWITCH,CHANGE
COMMON /SIG/SIG02,SIGC
COMMON /IUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLINT,KSTOR,NPLOTU
COMMON /DISCS/NDSCS,TDSC(20),XDSC(20),ZDSC(20),R2DSC(20),
1 QDSC(20,3)
COMMON /SPECS/ RES,STEP,TANT,CON
COMMON /CONST/PI,P12,PIRAD,TWOPI,TORRMB,CDEGK
EXTERNAL FUNCT,GFUN
DATA RES,TANT /.4,.1/
HORIZ=.TRUE.
ERR=.FALSE.
CON=ALOG(VISEXT)
CPTS5(2,1)=SPCHT
CPTS5(2,6)=SPCHT
U=U0+CPTS5(2,1)**UM

CLD00010
CLD00020
CLD00030
CLD00040
CLD00050
CLD00060
CLD00070
CLD00080
CLD00090
CLD00100
CLD00110
CLD00120
CLD00130
CLD00140
CLD00150
CLD00160
CLD00170
CLD00180
CLD00190
CLD00200
CLD00210
CLD00220
CLD00230
CLD00240
CLD00250
CLD00260
CLD00270
CLD00280
CLD00290
CLD00300
CLD00310
CLD00320
CLD00330
CLD00340
CLD00350
CLD00360
CLD00370
CLD00380
CLD00390
CLD00400
CLD00410
CLD00420
CLD00430
CLD00440
CLD00450
CLD00460
CLD00470
CLD00480
CLD00490
CLD00500
CLD00510
CLD00520
CLD00530
CLD00540
CLD00550
CLD00560
CLD00570
CLD00580
CLD00590
CLD00600
CLD00610
CLD00620
CLD00630
CLD00640
CLD00650
CLD00660
CLD00670
CLD00680
CLD00690
CLD00700

	CPTSS(1,1)=T*U*SINTH	CLD00710
	CPTSS(1,6)=CPTSS(1,1)	CLD00720
	NSERCH=-1	CLD00730
	STEP=20.	CLD00740
C	CALL CLIMB TO SEARCH FOR THE EDGE OF THE CLOUD IN ONE DIRECTION AT	CLD00750
C	THE HEIGHT OF THE OBSERVER.	CLD00760
C	CALL CLIMB(FUNCT,GFUN,CPTSS,FP1,NSERCH,NOCONT)	CLD00770
	NSERCH=1	CLD00780
	STEP=20.	CLD00790
C	CALL CLIMB TO SEARCH FOR THE EDGE OF THE CLOUD IN THE OPPOSITE	CLD00800
C	DIRECTION AT THE HEIGHT OF THE OBSERVER.	CLD00810
C	CALL CLIMB(FUNCT,GFUN,CPTSS(1,6),FP1,NSERCH,NOCONT)	CLD00820
	SPCWTH=CPTSS(1,6)-CPTSS(1,1)	CLD00830
	NCPTS=6	CLD00840
	IF(T.LE.TWIND)GO TO 50	CLD00850
	CNTRD(1)=(XTR+VTR*(T-TTR))*SINTH+CENDIF	CLD00860
	CNTRD(2)=ZTR	CLD00870
	SIGX2=SIG02+2.*KX*(T-TTR)	CLD00880
	SIGZ2=SIG02+2.*KZ*(T-TTR)	CLD00890
	SIGX=SQRT(SIGX2)	CLD00900
	SIGZ=SQRT(SIGZ2)	CLD00910
	BOT=1./2.*VISEXT)	CLD00920
	ARG=BOT*QPUFF(1)/PI/SIGX/SIGZ	CLD00930
	IF(ARG.LT.1.0)GO TO 998	CLD00940
	TOP(1)=CNTRD(1)	CLD00950
	TOP(2)=ZTR+SIGZ*SQRT(2.*ALOG(ARG))	CLD00960
	RAD=SIGX*SQRT(2.*ALOG(ARG))	CLD00970
	CENWTH=2.*RAD*PRJARY	CLD00980
	HEIGHT=TOP(2)	CLD00990
	GO TO 100	CLD01000
50	CNTRD(1)=XCM*SCRN(1)+YCM*SCRN(2)+CENDIF	CLD01010
	CNTRD(2)=ZCM	CLD01020
	TOP(1)=XTOP*SCRN(1)+YTOP*SCRN(2)+CENDIF	CLD01030
	TOP(2)=ZCM+RSPH	CLD01040
	IF(TOP(2).GT.ZINV)TOP(2)=ZINV	CLD01050
	HEIGHT=TOP(2)	CLD01060
	CENWTH=2.*(RSPH+PRJARY)	CLD01070
100	CPTSS(1,2)=CNTRD(1)-CENWTH/2.	CLD01080
	CPTSS(2,2)=CNTRD(2)	CLD01090
	CPTSS(1,3)=TOP(1)-PRJARY	CLD01100
	CPTSS(2,3)=TOP(2)	CLD01110
	CPTSS(1,4)=TOP(1)+PRJARY	CLD01120
	CPTSS(2,4)=TOP(2)	CLD01130
	CPTSS(1,5)=CNTRD(1)+CENWTH/2.	CLD01140
	CPTSS(2,5)=CNTRD(2)	CLD01150
	NCPTS=6	CLD01160
	GO TO 999	CLD01170
998	WRITE(1000,1000)	CLD01180
1000	FORMAT(50H *** UPPER PART OF CLOUD HAS DISSIPATED ***	CLD01190
999	RETURN	CLD01200
	END	CLD01210
		CLD01220
		CLD01230
		CLD01240
		CLD01250

```

      SUBROUTINE COMPCAL(NEWATM,NATMOS,ZTEMP,TMPMES,ZWIND,WNDMES,THWIND,
1      PHI,NEWSRC,CHWT,NCHRG,NCHS,DETDEP,NSOIL,DSOD,
2      NWL,TRNCOR,RECCOR,TIME,DHDT,TRNLOS,NERR)
      CONTROLLING ROUTINE FOR PRECOMPUTING A SINGLE , EXPLOSION PRODUCED
      DUST CLOUD AND STORING IT ON AN EXTERNAL FILE TO BE USED AT A
      LATER RUNNING OF THE CODE FOR A RANDOM DISTRIBUTION IN SPACE AND TIME
      OF CHARGES
      INPUTS
      FOR DETAILS SEE DRTAN
      OUTPUTS
      TRNLOS -CALCULATE TRANSMITTANCE ALONG THE SPECIFIED LINE OF SIGHT
      *****
      DIMENSION ZTEMP(2),TMPMES(2),ZWIND(2),WNDMES(2),TRNCOR(3),
1 RECCOR(3),TRNFRM(2,2),SIDE1(2),SIDE2(2),NCHS(2),ORIG(2),TRN(3),
2 REC(3),SRCBAS(2),PAS(6)
      LOGICAL DHDT,NEWATM,NEWSRC,ERR,FLAG
      COMMON/GEOM/COSTH2,SINTH,SINTH2,VISEXT,RTP1,SCRN(2)
      COMMON/IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
      COMMON/VL/VLOAD
      COMMON/OPTION/IOPT,IFILE
      COMMON/CLOCK/FTIME,TWIND
      COMMON/BUOYCL/Y(8),SPHNS(3),RTIM
      COMMON/DISC/NDSCS,TDSC(20),XDSC(20),ZDSC(20),R2DSC(20),QDSC(20,3)
      COMMON/M05/DIFF(2,200),NCHTOT,PRSEP(200),NTOT,NARY,ITOT,
+ COOR(2,200),TSTAG(200),
+ ICOUNT,TIMES(25),XC0(3,25),XC1(3,25),RT(3,25),
1 RB(3,25),Z2(3,25)
      COMMON/WNDPRM/DXZ0,DYZ0,DZ0,U0,ON,ZINV
      COMMON/CARB/RCARB1,RCARB2
      COMMON/CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK
      DATA FLAG/.TRUE./
      DATA PAS/4HA,4HB,4HC,4HD,4HE,4HF /
      IF(IOPT.EQ.4)GO TO 500
      PRECOMPUTE INFORMATION AND STORE ON FORTRAN UNIT IFILE
      THETAX=THWIND*PI/180.
      TWIND=1.E5
      TTR=1.E5
      IF(.NOT.NEWATM)GO TO 10
      CALL ATMCAL(NATMOS,ZTEMP,TMPMES,ZWIND,WNDMES,PHI,THETAX,DHDT,ERR)
      IF(.NOT.ERR)GO TO 10
      NERR=7
      GO TO 999
10 CONTINUE
      IF(.NOT.NEWSRC)GO TO 20
      CALL SOURCE(CHWT,NCHRG,DETDEP,NSOIL,DSOD)
20 CONTINUE
      COMPUTE INITIAL LOADING
      SUM=0.0
      DO 25 I=1,NDSCS
      SUM=SUM+QDSC(I,1)
25 CONTINUE
      VLOAD=SPHNS(1)+SUM
      CALL PRECL TO COMPUTE AND STORE THE QUADRATIC FITS NECESSARY
      FOR THE CONE
      CALL PRECL(NATMOS,ZTEMP,TMPMES,ZWIND,WNDMES,THWIND,PHI,DHDT,
1 CHWT,NCHRG,DETDEP,NSOIL,DSOD,SILT)
      GO TO 999
500 CONTINUE

```

```

      IF(.NOT.FLAG) GO TO 35
      READ(IFILE)NATMOS,ZTEMP(1),TMPMES(1),ZWND(1),WNDMES(1)
      READ(IFILE)DHDT,PHI,CHWT,NCHRG,DETDEP,NSOIL,DSOD,SILT,ZINV
      READ(IFILE)YLOAD,RCARB1,RCARB2
      READ(IFILE)ICOUNT
      DO 30 J=1,ICOUNT
      READ(IFILE) TIMES(J),(RT(I,J),RB(I,J),Z2(I,J),XC0(I,J),
1      XC1(I,J),I=1,3)
      30 CONTINUE
      WRITE(IOOUT,800)
800 FORMAT(/,5X,85H ATMOSPHERIC, CHARGE, AND SOIL CHARACTERISTICS USED
1 WHEN THE CLOUD WAS PRECOMPUTED.)
C C C C C
      WRITE OUT ATMOSPHERIC INFORMATION
      WRITE(IOOUT,808)
808 FORMAT(1X)
      WRITE(IOOUT,810)PAS(NATMOS)
      810 FORMAT(30H PASQUILL CATEGORY ,A4)
      NIO=1
      WRITE(IOOUT,812)(ZTEMP(I),TMPMES(I),ZWND(I),WNDMES(I),I=1,NIO)
      812 FORMAT(8H HT ,F8.2,7H TEMP ,F8.2,7H HT ,F8.2,7H WIND ,
1 F8.2)
      WRITE(IOOUT,814)THWND
      814 FORMAT(22H WIND DIRECTION ,F8.2)
      WRITE(IOOUT,819)PHI
      IF(DHDT)WRITE(IOOUT,816)
      816 FORMAT(40H THE INVERSION LAYER HEIGHT IS GROWING )
      IF(.NOT.DHDT)WRITE(IOOUT,818)
      818 FORMAT(42H THE INVERSION LAYER HEIGHT IS CONSTANT )
      819 FORMAT(22H LATITUDE ,F8.2)
C C C C C
      WRITE SOIL CHARACTERISTICS
      IF(NSOIL.EQ.1)WRITE(IOOUT,821)
      820 FORMAT(15H SILT CONTENT ,F5.2)
      821 FORMAT(/,15H SOIL-1 )
      IF(NSOIL.EQ.2)WRITE(IOOUT,822)
      822 FORMAT(/,15H SOIL-2 )
      WRITE(IOOUT,820)SILT
      WRITE(IOOUT,823)DSOD
      823 FORMAT(21H DEPTH OF SOD ,F5.2)
C C C C C
      WRITE EXPLOSIVE CHARGE CHARACTERISTICS
      IF(NCHRG.EQ.1)WRITE(IOOUT,824)
      824 FORMAT(/,65H SURFACE - LIVE FIRE OR 30 DEGREE TILTED STATIC, TIP
1 ON GROUND )
      IF(NCHRG.EQ.2)WRITE(IOOUT,825)
      825 FORMAT(/,25H BARE CHARGE ON SURFACE )
      IF(NCHRG.EQ.3)WRITE(IOOUT,826)
      826 FORMAT(/,45H 30 DEGREE TILTED TIP AT 0.3 METER DEPTH )
      IF(NCHRG.EQ.4)WRITE(IOOUT,827)
      827 FORMAT(/,45H 30 DEGREE TILTED TIP AT 0.6 METER DEPTH )
      IF(NCHRG.EQ.5)WRITE(IOOUT,828)
      828 FORMAT(/,40H HORIZONTAL PROJECTILE ON SURFACE )
      WRITE(IOOUT,829)CHWT
      829 FORMAT(30H WEIGHT OF CHARGE ,F8.2,5H KG. )
      WRITE(IOOUT,830)DETDEP
      830 FORMAT(30H DETONATION DEPTH ,F8.2)
      FLAG=.FALSE.
C C C C C
      COMPUTE THE ROTATION TRANSFORMATION MATRIX TO CONVERT THE USER
      C DEFINED COORDINATES INTO LOCAL COORDINATES WITH THE X-AXIS IN
      C THE WIND DIRECTION.
      35 CONTINUE
      IF(.NOT.NEWATM) GO TO 45
      THETA=THWND*PI/180.
      TRNFRM(1,1)=COS(THETA)

```

```

COM00700
COM00710
COM00720
COM00730
COM00740
COM00750
COM00760
COM00770
COM00780
COM00790
COM00800
COM00810
COM00820
COM00830
COM00840
COM00850
COM00860
COM00870
COM00880
COM00890
COM00900
COM00910
COM00920
COM00930
COM00940
COM00950
COM00960
COM00970
COM00980
COM00990
COM01000
COM01010
COM01020
COM01030
COM01040
COM01050
COM01060
COM01070
COM01080
COM01090
COM01100
COM01110
COM01120
COM01130
COM01140
COM01150
COM01160
COM01170
COM01180
COM01190
COM01200
COM01210
COM01220
COM01230
COM01240
COM01250
COM01260
COM01270
COM01280
COM01290
COM01300
COM01310
COM01320
COM01330
COM01340
COM01350
COM01360
COM01370

```

	TRNFRM(2,2)=TRNFRM(1,1)	COM01380
	TRNFRM(1,2)=SIN(THETA)	COM01390
	TRNFRM(2,1)=-TRNFRM(1,2)	COM01400
C		COM01410
C	MAKE THE ORIGIN OF THE LOCAL COORDINAT SYSTEM THE FIRST	COM01420
C	CHARGE LOCATION THAT WAS INPUT BY THE USER	COM01430
	DO 40 I=1,2	COM01440
	SRCBAS(I)=COORD(I,1)	COM01450
	ORIG(I)=SRCBAS(I)	COM01460
	40 CONTINUE	COM01470
C		COM01480
C	CALL SETUP TO COMPUTE THE ARRAY OF DIFFERENCE VECTORS	COM01490
C	CALL SETUP(NCHS,SRCBAS,SIDE1,SIDE2,TRNFRM)	COM01500
C		COM01510
C	COMPUTE COORDINATES OF THE TRANSMITTER AND RECEIVER IN THE LOCAL	COM01520
C	COORDINATE SYSTEM	COM01530
	TRN(3)=TRNCOR(3)	COM01540
	REC(3)=RECCOR(3)	COM01550
	DO 60 I=1,2	COM01560
	TRN(I)=0.0	COM01570
	REC(I)=0.0	COM01580
	DO 50 J=1,2	COM01590
	TRN(I)=TRN(I)+TRNFRM(I,J)*(TRNCOR(J)-ORIG(J))	COM01600
	REC(I)=REC(I)+TRNFRM(I,J)*(RECCOR(J)-ORIG(J))	COM01610
	50 CONTINUE	COM01620
	60 CONTINUE	COM01630
	CONTINUE	COM01640
45		COM01650
C	CALL PRETRN TO COMPUTE THE TRANSMITTANCE ALONG THE SPECIFIED LINE OF	COM01660
C	SIGHT	COM01670
	CALL PRETRN(TRN,REC,TIME,TRNLOS)	COM01680
	999 RETURN	COM01690
	END	COM01700
		COM01710
		COM01720
		COM01730

```

SUBROUTINE CONLEN(U,TR,HTTOP,HTBOT,XCEN,YCEN,RTOP,RBOT,XB,YB,
1XNORM,PLEN)
ROUTINE TO FIND THE LENGTH OF A NON-HORIZONTAL LINE THAT INTERSECTS
A CONE
INPUTS
    U      - UNIT VECTOR ALONG LINE CONNECTING THE TRANSMITTER AND
            RECEIVER.
    TR      - TRANSMITTER COORDINATES
    HTTOP-  HEIGHT OF THE TOP OF THE CONE SHAPED PORTION OF THE CLOUD
    HTBOT-  HEIGHT OF THE BOTTOM OF THE CONE SHAPED PORTION OF THE
            CLOUD
    XCEN - X POSITION OF THE CENTER OF THE CONE SHAPED CLOUD AT TOP
    YCEN - Y POSITION OF THE CENTER OF THE CONE SHAPED CLOUD AT TOP
    RTOP - RADIUS OF THE CONE AT THE TOP
    RBOT - RADIUS OF THE CONE AT THE BOTTOM
    XB    - X POSITION OF THE BOTTOM OF THE CONE SHAPED CLOUD
    YB    - Y POSITION OF THE BOTTOM OF THE CONE SHAPED CLOUD
OUTPUT
    PLEN - LENGTH OF THE INTERSECTION OF CONE AND THE LINE OF SIGHT
FUNCTIONS AND SUBROUTINES
    NONE
*****
C *****
C DIMENSION U(3),TR(3)
C IF(U(3).LT.0.0)GO TO 40
C SET UP BOUNDS SO INTERSECTION OF LINE IS SUCH THAT HTBOT < Z < HTTOP
C
C   PMIN=(HTBOT-TR(3))/U(3)
C   PMAX=(HTTOP-TR(3))/U(3)
C   GO TO 50
40 PMIN=(HTTOP-TR(3))/U(3)
C   PMAX=(HTBOT-TR(3))/U(3)
50 P1=(HTTOP-HTBOT)/U(3)
C   P0=(HTBOT-TR(3))/U(3)
C SET UP QUADRATIC TO BE SOLVED
C
C   DX1=U(1)*P1-XCEN+XB
C   DY1=U(2)*P1-YCEN+YB
C   DR=RTOP-RBOT
C   A=DX1**2+DY1**2-DR**2
C   DX0=TR(1)+U(1)*P0-XB
C   DY0=TR(2)+U(2)*P0-YB
C   B=2.*(DX1*DX0+DY1*DY0-DR*RBOT)
C   C=(DX0**2+DY0**2-RBOT**2)
C   RADIC1=(DX1*RBOT-DX0*DR)**2
C   RADIC2=(DY1*RBOT-DY0*DR)**2
C   RADIC3=(DX1*DY0-DY1*DX0)**2
C DETERMINE PATH LENGTH IF THE LINE INTERSECTS THE CONE
C
C   IF(ABS(A).LT.1.E-20)GO TO 60
C   RADIC=RADIC1+RADIC2-RADIC3

```

```

CNL00380
CNL00390
CNL00010
CNL00020
CNL00030
CNL00040
CNL00050
CNL00060
CNL00070
CNL00080
CNL00090
CNL00100
CNL00110
CNL00120
CNL00130
CNL00140
CNL00150
CNL00160
CNL00170
CNL00180
CNL00190
CNL00200
CNL00210
CNL00220
CNL00230
CNL00240
CNL00250
CNL00260
CNL00270
CNL00280
CNL00290
CNL00300
CNL00310
CNL00320
CNL00330
CNL00340
CNL00350
CNL00360
CNL00370
CNL00400
CNL00410
CNL00420
CNL00430
CNL00440
CNL00450
CNL00460
CNL00470
CNL00480
CNL00490
CNL00500
CNL00510
CNL00520
CNL00530
CNL00540
CNL00550
CNL00560
CNL00570
CNL00580
CNL00590
CNL00600
CNL00610
CNL00620
CNL00630
CNL00640
CNL00650
CNL00660
CNL00670
CNL00680
CNL00690
CNL00700

```

	IF(RADIC.LT.0.0)GO TO 68	CNL00710
	ROOT1=-B/A/2.+SQRT(RADIC)/A	CNL00720
	ROOT2=-B/A/2.-SQRT(RADIC)/A	CNL00730
C		CNL00740
C	DETERMINE WHICH POINT IS CLOSEST AND WHICH IS FARTHEST FROM THE	CNL00750
C	TRANSMITTER	CNL00760
	TEMP1=P1*ROOT1+P0	CNL00770
	TEMP2=P1*ROOT2+P0	CNL00780
	IF (A.LT.0.0) GO TO 57	CNL00790
	IF(TEMP1.GT.TEMP2)GO TO 55	CNL00800
	PFAR=TEMP2	CNL00810
	PNEAR=TEMP1	CNL00820
	GO TO 67	CNL00830
55	PFAR=TEMP1	CNL00840
	PNEAR=TEMP2	CNL00850
	GO TO 67	CNL00860
57	IF (U<3).GT.0.0) GO TO 58	CNL00870
	PNEAR=-1.E20	CNL00880
	PFAR=AMIN1(TEMP1,TEMP2)	CNL00890
	GO TO 67	CNL00900
58	PNEAR=AMAX1(TEMP1,TEMP2)	CNL00910
	PFAR=1.E20	CNL00920
	GO TO 67	CNL00930
C		CNL00940
C	CONSIDER DEGENERATE CASE	CNL00950
C		CNL00960
60	IF(B.LT.1.E-20)GO TO 68	CNL00970
	IF(P1/B.LT.0.0)GO TO 65	CNL00980
	PFAR=P1*(-C/B)+P0	CNL00990
	PNEAR=0.0	CNL01000
	GO TO 67	CNL01010
65	PFAR=XNORM	CNL01020
	PNEAR=P1*(-C/B)+P0	CNL01030
67	PLEN=AMIN1(PFAR,XNORM,PMAX)-AMAX1(PNEAR,0.0,PMIN)	CNL01040
	IF(PLEN.LT.0.0)PLEN=0.0	CNL01050
	GO TO 999	CNL01060
68	PLEN=0.0	CNL01070
999	RETURN	CNL01080
	END	CNL01090
		CNL01100

998 QCOL=WAKAL/4./SQRT(PI)
RETURN
END

COV00700
COV00710
COV00720

C
C
C

XC=XTR+VTR*(T-TTR)+DIFF(1,NCHG)	CSP00710
YC=DIFF(2,NCHG)	CSP00720
ARG1=((X-XC)**2+(Y-YC)**2)/2./SIGX2	CSP00730
IF(ABS(ARG1).GT.30.)GO TO 999	CSP00740
TERM=SPHAL/((2.*PI)**(3./2.))/SIGZ/SIGX2	CSP00750
CWINDC=TERM*EXP(-ARG1)*(TERM2+TERM3)	CSP00760
GO TO 100	CSP00770
COMPUTE COLUMN DENSITY	CSP00780
50 TERM=SPHAL/2./PI/SIGX/SIGZ	CSP00790
DO 90 I=1,NCHTOT	CSP00800
XC=XTR+VTR*(T-TTR)+DIFF(1,I)	CSP00810
YC=DIFF(2,I)	CSP00820
XY=XC*SCRN(1)+YC*SCRN(2)	CSP00830
ARG1=(X-XY)**2/2./SIGX2	CSP00840
IF(ABS(ARG1).GT.30.)GO TO 90	CSP00850
CWINDSC=TERM*EXP(-ARG1)*(TERM2+TERM3)	CSP00860
CWINDC=CWINDC+CWINDSC	CSP00870
CALL TRNCHK(CWINDS,CWINDW,CWINDC)	CSP00880
IF(TEST)GO TO 999	CSP00890
90 CONTINUE	CSP00900
100 CSPHER=CWINDC	CSP00910
999 RETURN	CSP00920
END	CSP00930
	CSP00940
	CSP00950

```

FUNCTION CWAKE(X,Y,Z,T)
FUNCTION TO COMPUTE THE LOCAL CONCENTRATION OR COLUMN DENSITY AT
X,Y,Z AND TIME T AFTER THE BLAST FOR THE WAKE.

INPUT
  X,Y,Z      COORDINATES IN METERS. IF THE LINE INTEGRAL IS DESIRED
              THESE ARE NOT USED AND THE LINE IS SPECIFIED BY THE
              TRANSMITTER AND RECEIVER COORDINATES AND INFORMATION
              CALCULATED AT THE TIME THE BUOYANT FIREBALL CONVERTED
              TO THE WIND MODEL.
  T          THE TIME IN SECONDS AFTER DETONATION

OUTPUT
  RETURNS THE CONCENTRATION AT X,Y,Z,T IF HORIZ IS .FALSE. AND
  THE LINE INTEGRAL OF CONCENTRATION (COLUMN DENSITY) IF
  HORIZ IS .TRUE.

FUNCTIONS AND SUBROUTINES NEEDED
  ERF        COMPUTE THE ERROR FUNCTION

*****
REAL M,N,KZ,KX
LOGICAL HORIZ, SWITCH, CHANGE, TEST
COMMON/PRTIME/ R0,VGRAV(3),NPRTS
COMMON/GEOM/COSTH2,SINTH,SINTH2,VISEXT,RTPI,SCRN(2)
COMMON/MODE/ HORIZ
COMMON/WINDPRM/DXZ0,DYX0,DZ0,U0,M,N,ZINV
COMMON/DISCS/NDSCS,TDSC(20),XDSC(20),ZDSC(20),R2DSC(20),
1 QDSC(20,3)
COMMON/M05/DIFF(2,200),NCHTOT,PRSEP(200),NTOT,NARY,ITOT,
+ COOR(2,200),TSTAG(200),DMMY(401)
COMMON/TRAN/VTR,KZ,KX,TTR,XTR,ZTR,QPUFF(3),SWITCH,CHANGE
COMMON/SIG/SIG02,SIGC
COMMON/IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLINT,KSTOR,NPLOTUC
COMMON/LOAD/WAKAL,SPHAL
COMMON/ACL/CWINDS,CWINDC,CWINDW
COMMON/WAKE/XDIF,YDIF,ZDIF,TDIF,TDX,TDZ,QLOC,QCOL,XBAYRG
COMMON/LOS/TR(3),RE(3),U(3)
COMMON/CHARGE/NCHG
COMMON/TRANNNY/THRESH,TEST,NWL,NSOIL
COMMON/CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDECK
IF(NARY.EQ.3)GO TO 999
IF((T-TTR).LT.1.E-20)GO TO 999

COMPUTE CONTRIBUTION FROM THE WAKE AFTER SWITCHING TO THE WIND MODEL
FOR A SINGLE CHARGE
  CWAKE=0.0
  CWINDW=0.0
  CWINDSW=0.0
  SIGX2=TDX*(TDIF+(T-TTR))
  SIGZ2=TDZ*(T-TTR)
  IF(HORIZ)GO TO 210
  XB=XBAYRG+DIFF(1,NCHG)
  YB=DIFF(2,NCHG)

COMPUTE THE LOCAL CONCENTRATION
  A=-(XDIF**2+YDIF**2)/2./SIGX2-(ZDIF**2/2./SIGZ2)
  XX=X-VTR*(T-TTR)
  B0=(XDIF*(XX-XB)+YDIF*(Y-YB))/SIGX2
  B1=B0+(ZDIF*(Z-5.)/SIGZ2)

```

B2=B0-(ZDIF*(Z+5.)/SIGZ2)	CWA00710
C0=-(XX-XB)**2+(Y-YB)**2)/2./SIGX2	CWA00720
C1=C0-(Z-5.)/SIGZ2	CWA00730
C2=C0-(Z+5.)/SIGZ2	CWA00740
RTMA=SQRT(-A)	CWA00750
ARG1=C1-(B1**2/4./A)	CWA00760
ARG2=-B1/2./RTMA	CWA00770
ARG3=RTMA+ARG2	CWA00780
ARG4=C2-(B2**2/4./A)	CWA00790
ARG5=-B2/2./RTMA	CWA00800
ARG6=RTMA+ARG5	CWA00810
IF(ARG1.LT.-30.)GO TO 221	CWA00820
TERM1=EXP(ARG1)*(ERF(ARG3)-ERF(ARG2))	CWA00830
GO TO 222	CWA00840
221 TERM1=0.0	CWA00850
222 CONTINUE	CWA00860
IF(ARG4.LT.-30.)GO TO 223	CWA00870
TERM2=EXP(ARG4)*(ERF(ARG6)-ERF(ARG5))	CWA00880
GO TO 224	CWA00890
223 TERM2=0.0	CWA00900
224 CWAKE=QLOC/SIGX2/SQRT(SIGZ2)/RTMA*(TERM1+TERM2)	CWA00910
GO TO 999	CWA00920
CC COMPUTE COLUMN DENSITY	CWA00930
210 DO 245 J=1,NCHTOT	CWA00940
XB=XBAVRG+DIFF(1,J)	CWA00950
YB=DIFF(2,J)	CWA00960
230 A=-(XDIF*U(1)-YDIF*U(2))**2/2./SIGX2-(ZDIF)**2/2./SIGZ2	CWA00970
TRR=TR(1)-VTR*(T-TTR)	CWA00980
B0=(YDIF*U(1)-XDIF*U(2))*(TR(2)-YB)*U(1)-(TRR-XB)*U(2))/SIGX2	CWA00990
B1=B0+ZDIF*(TR(3)-5.)/SIGZ2	CWA01000
B2=B0-ZDIF*(TR(3)+5.)/SIGZ2	CWA01010
C0=-(TR(2)-YB)*U(1)-(TRR-XB)*U(2))**2/2./SIGX2	CWA01020
C1=C0-(TR(3)-5.)/SIGZ2	CWA01030
C2=C0-(TR(3)+5.)/SIGZ2	CWA01040
RTMA=SQRT(-A)	CWA01050
ARG1=C1-B1**2/4./A	CWA01060
ARG2=-B1/2./RTMA	CWA01070
ARG3=RTMA+ARG2	CWA01080
ARG4=C2-B2**2/4./A	CWA01090
ARG5=-B2/2./RTMA	CWA01100
ARG6=RTMA+ARG5	CWA01110
IF(ARG1.GT.30.)ARG1=30.	CWA01120
IF(ARG1.LT.-30.)GO TO 231	CWA01130
TERM1=EXP(ARG1)*(ERF(ARG3)-ERF(ARG2))	CWA01140
GO TO 232	CWA01150
231 TERM1=0.0	CWA01160
232 CONTINUE	CWA01170
IF(ARG4.LT.-30.)GO TO 233	CWA01180
TERM2=EXP(ARG4)*(ERF(ARG6)-ERF(ARG5))	CWA01190
GO TO 234	CWA01200
233 TERM2=0.0	CWA01210
234 ARG=SIGX2*S'GZ2	CWA01220
CWNSW=(QCOL/SQRT(ARG)/RTMA)*(TERM1+TERM2)	CWA01230
240 CONTINUE	CWA01240
CWINDW=CWINDW+CWNSW	CWA01250
CALL TRNCHK(CWINDS,CWINDW,CWINDC)	CWA01260
IF(TEST)GO TO 999	CWA01270
245 CONTINUE	CWA01280
CWAKE=CWINDW	CWA01290
999 RETURN	CWA01300
END	CWA01310
	CWA01320
	CWA01330

FUNCTION CWIND(X,Y,Z,T)

PURPOSE

TO COMPUTE THE CONCENTRATION AT A POINT OR INTEGRATED ALONG
A HORIZONTAL LINE (SKIRT PORTION OF DUST CLOUD)

INPUT

X,Y,Z COORDINATES IN METERS. IF LINE INTEGRAL IS DESIRED,
Y IS IGNORED AND LINE IS SPECIFIED BY X AND Z.

T THE TIME IN SECONDS AFTER DETONATION

OUTPUT

RETURNS THE CONCENTRATION AT X,Y,Z,T IF HORIZ IS .FALSE. AND
THE LINE INTEGRAL OF CONCENTRATION IF HORIZ IS .TRUE.

SUBROUTINES CALLED

MOMENT COMPUTES ZERO ORDER MOMENT AND INTERPOLATES FROM
TABLE OF HIGHER ORDER MOMENTS.

CALLED BY FUNCT,TRNCAL

REAL M,N,KZ,KX
LOGICAL HORIZ, SWITCH, CHANGE, TEST, SKIP
DIMENSION REF(2), REF0(2)
COMMON /PRTINF/ R0, VGRAV(3), NPRTS
COMMON /GEOM/ COSTH2, SINTH, SINTH2, VISEXT, RTP1, SCRN(2)
COMMON /MODE/ HORIZ
COMMON /WINDPRM/ DX20, DYX0, DZ0, U0, M, N, ZINV
COMMON /DISCS/ NDSCS, TDSC(20), XDSC(20), ZDSC(20),
1 QDSC(20,3)
COMMON /MO5/ DIFF(2,200), NCHTOT, PRSEP(200), NTOT, NARY, ITOT,
+ COOR(2,200), TSTAG(200), DMMY(401)
COMMON /TRAN/ VTR, KZ, KX, TTR, XTR, ZTR, QPUFF(3), SWITCH, CHANGE
COMMON /SIG/ SIG02, SIGC
COMMON /IOUNIT/ IOIN, IOOUT, IPHFUN, LOUNIT, NDIRTU, NCLIMT, KSTOR, NPLOTUC
COMMON /LOAD/ WAKAL, SPHAL
COMMON /ACL/ CWINDS, CWINDC, CWINDW
COMMON /WAKE/ XDIF, YDIF, ZDIF, TDIF, TDX, TDZ, QLOC, QCOL, XBAYRG
COMMON /LOS/ TR(3), RE(3), U(3)
COMMON /CHARGE/ NCHG
COMMON /TRANNY/ THRESH, TEST, NWL, NSOIL
COMMON /SKIPIT/ SKIP
COMMON /CONST/ PI, PI2, PIRAD, TWOPI, TORRMB, CDEGK

COMMON /PRTINF/

R0 INITIAL RADIUS OF THE CLOUD IN METERS
VGRAV SINGLY DIMENSIONED ARRAY. VGRAV(I) IS THE OPTICALLY
WEIGHTED AVERAGE SETTLING VELOCITY FOR PARTICLES IN THE
I SIZE RANGE
NPRTS THE NUMBER OF PARTICLE SIZE RANGES

COMMON /DISCS/

NDSCS THE NUMBER OF DISC SOURCES
TDSC SINGLY DIMENSIONED ARRAY CONTAINING THE TIME OF RELEASE
OF THE DISC SOURCES
XDSC SINGLY DIMENSIONED ARRAY CONTAINING THE X COORDINATE
OF THE CENTER OF THE DISC SOURCES
ZDSC SINGLY DIMENSIONED ARRAY CONTAINING THE Z COORDINATE
OF THE CENTER OF THE DISC SOURCES
R2DSC SINGLY DIMENSIONED ARRAY CONTAINING THE SQUARE OF THE
RADIUS OF THE DISC SOURCES
QDSC DOUBLY DIMENSIONED ARRAY. QDSC(I,J) IS THE OPTICALLY

CWI00270
CWI00010
CWI000020
CWI000030
CWI000040
CWI000050
CWI000060
CWI000070
CWI000080
CWI000090
CWI000100
CWI000110
CWI000120
CWI000130
CWI000140
CWI000150
CWI000160
CWI000170
CWI000180
CWI000190
CWI000200
CWI000210
CWI000220
CWI000230
CWI000240
CWI000250
CWI000260
CWI000280
CWI000290
CWI000300
CWI000310
CWI000320
CWI000330
CWI000340
CWI000350
CWI000360
CWI000370
CWI000380
CWI000390
CWI000400
CWI000410
CWI000420
CWI000430
CWI000440
CWI000450
CWI000460
CWI000470
CWI000480
CWI000490
CWI000500
CWI000510
CWI000520
CWI000530
CWI000540
CWI000550
CWI000560
CWI000570
CWI000580
CWI000590
CWI000600
CWI000610
CWI000620
CWI000630
CWI000640
CWI000650
CWI000660
CWI000670
CWI000680
CWI000690
CWI000700

C	WEIGHTED MASS OF PARTICLES OF THE J SIZE RANGE IN THE	CW100710
C	1 DISC.	CW100720
C		CW100730
C		CW100740
C	SUM THE CONTRIBUTIONS OF THE DISC SOURCES TO THE	CW100750
C	OPTICALLY WEIGHTED CONCENTRATION AT (X,Y,Z,T)	CW100760
	CWIND=0.	CW100770
	CWINDSC=0.0	CW100780
	CWINDS=0.0	CW100790
	CWINDC=0.0	CW100800
	CWINDW=0.0	CW100810
	IF(HORIZ)GO TO 110	CW100820
C		CW100830
C	COMPUTE CONCENTRATION AT X,Y,Z (FOR SIMULTANEOUS BURST)	CW100840
C		CW100850
	DO 100 I=1,NDSCS	CW100860
	REF0(1)=XDSC(I)	CW100870
	REF0(2)=0.0	CW100880
	ROH2=R2DSC(I)	CW100890
	H=ZDSC(I)	CW100900
	TOF=T-TDSC(I)	CW100910
	DO 90 J=1,NPRTS	CW100920
C		CW100930
C	DETERMINE MOMENTS FOR CURRENT SOURCE DISC AT Z	CW100940
C		CW100950
	CALL MOMENT(VGRAV(J),Z,H,TOF,Q,XBAR,SIGW2,SIGP2)	CW100960
	IF(Q.GT.1.E-10)GO TO 50	CW100970
	CWINDSC=0.0	CW100980
	GO TO 100	CW100990
50	CONTINUE	CW101000
	RX2=ROH2+2.*SIGW2	CW101010
	RY2=ROH2+2.*SIGP2	CW101020
	DO 114 NA=1,2	CW101030
	REF(NA)=REF0(NA)+DIFF(NA,NCHG)	CW101040
114	CONTINUE	CW101050
	ARG=-(X-REF(1)-XBAR)**2/RX2	CW101060
	IF(ABS(ARG).GT.30.)GO TO 100	CW101070
	CWINDSC=(Q/RTPI/SQRT(RX2))*EXP(ARG)	CW101080
	ARG=-(Y-REF(2))**2/RY2	CW101090
	IF(ABS(ARG).GT.30.)GO TO 100	CW101100
	CY=EXP(ARG)/RTPI/SQRT(RY2)	CW101110
	CWINDSC=CWINDSC(I,J)*CWINDSC*CY	CW101120
	CWINDS=CWINDS+CWINDSC	CW101130
90	CONTINUE	CW101140
100	CONTINUE	CW101150
	CWIND=CWINDS	CW101160
	GO TO 999	CW101170
110	DO 220 ICHG=1,NTOT	CW101180
	IF(T.LT.TSTAG(ICHG))GO TO 220	CW101190
	DO 211 I=1,NDSCS	CW101200
	TOF=T-TDSC(I)-TSTAG(ICHG)	CW101210
	REF0(1)=XDSC(I)	CW101220
	REF0(2)=0.	CW101230
	ROH2=R2DSC(I)	CW101240
	H=ZDSC(I)	CW101250
	IF(HORIZ) REF0(1)=REF0(1)*SINTH	CW101260
	DO 210 J=1,NPRTS	CW101270
	CWINDSC=0.0	CW101280
C		CW101290
C	DETERMINE MOMENTS FOR CURRENT SOURCE DISC AT Z	CW101300
C		CW101310
	CALL MOMENT(VGRAV(J),Z,H,TOF,Q,XBAR,SIGW2,SIGP2)	CW101320
	IF(Q.GT.1.E-10) GO TO 113	CW101330
C		CW101340
C	IF Q IS TOO SMALL, ITS CONTRIBUTION IS IGNORED	CW101350
C		CW101360
	CWINDSC=0.	CW101370
	GO TO 210	CW101380
113	CONTINUE	CW101390
		CW101400

	DO 200 MA=1,NCHTOT	CWI01410
	INDX=MA	CWI01420
	IF(NARY.EQ.3)INDX=ICMG	CWI01430
	RX2=R0H2+2.*SIGW2	CWI01440
	RY2=R0H2+2.*SIGP2	CWI01450
C	COMPUTE CONCENTRATION ALONG LINE OF SIGHT SPECIFIED BY X,Z	CWI01460
C		CWI01470
C		CWI01480
120	CONTINUE	CWI01490
	REF(1)=REF0(1)+PRSEP(INDX)	CWI01500
	REFF2=RX2*SINTH2+RY2*COSTH2	CWI01510
	ARG=-(X-REF(1)-XBAR*SINTH)**2/REFF2	CWI01520
	IF(ABS(ARG).GT.30.)GO TO 150	CWI01530
	CWINDSC=EXP(ARG) /SQRT(REFF2)/RTPI	CWI01540
	CWINDSC=CWINDSC*Q*QDSC(I,J)	CWI01550
150	CONTINUE	CWI01560
	CWINDS=CWINDS+CWINDSC	CWI01570
	IF(SKIP)GO TO 190	CWI01580
	CALL TRNCHK(CWINDS,CWINDW,CWINDC)	CWI01590
	IF(TEST)GO TO 999	CWI01600
190	CONTINUE	CWI01610
200	CONTINUE	CWI01620
210	CONTINUE	CWI01630
211	CONTINUE	CWI01640
220	CONTINUE	CWI01650
	CWIND=CWINDS	CWI01660
999	RETURN	CWI01670
	END	CWI01680

```

SUBROUTINE DIFEQ(N,T,Y,YP)
REAL KM,KZ,KX
LOGICAL SWITCH,CHANGE
DIMENSION Y(N),YP(N)
COMMON/PRTIME/RCL,VGRAV(3),NPRTS
COMMON/WNDPRM/DXZ0,DYX0,DZ0,U0,UM,ON,ZINV
COMMON /ARRAY/OVLAP,AREA,PERIM,PRJARY,CENDIF
COMMON/MOS/DIFF(2,200),NCHTOT,PRSEP(200),NTOT,NARY,ITOT,
+ DMM(600),DMY(401)
COMMON /BURST/ ACCEL,TBURST
COMMON/STARS/USTAR,TSTAR,ZSTAR
COMMON/EKWIND/ALP,C,PYF,PF,UHAT,VHAT
COMMON/EKTEMP/Z0,ZL,T0,TC1,TC2,TC3
COMMON/TRAN/VTR,KZ,KX,TTR,XTR,ZTR,QPUFF(3),SWITCH,CHANGE
COMMON/SIG/SIG02,SIGC
COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
DATA ALPHA/.25/
*****

PURPOSE
DIFEQ CONTAINS THE PARTIAL DIFFERENTIAL EQUATIONS FOR THE
RISE OF A BUOYANT CLOUD WHICH ARE USED BY SUBROUTINE RKM.

INPUT
N      THE NUMBER OF DEPENDENT VARIABLES
T      THE INDEPENDENT VARIABLE, I.E. TIME
Y(1)   RADIUS OF CLOUD
Y(2)   CLOUD TEMPERATURE MINUS SURROUNDING TEMPERATURE
Y(3)   VERTICAL VELOCITY OF CLOUD
Y(4)   X-COORDINATE OF CENTER OF MASS FOR THE CLOUD
Y(5)   Y-COORDINATE OF CENTER OF MASS FOR THE CLOUD
Y(6)   THE HEIGHT OF THE CLOUD C.O.M.
Y(7)   X-COORDINATE OF TOP OF CLOUD
Y(8)   Y-COORDINATE OF TOP OF CLOUD

OUTPUT
YP     AN ARRAY CONTAINING COMPUTED DERIVATIVES OF THE DEPENDENT
       VARIABLES WITH RESPECT TO THE INDEPENDENT VARIABLE.

REQUIRED FUNCTIONS
TEMP    CALCULATES AMBIENT ATMOSPHERIC TEMPERATURE AND THE
        TEMPERATURE GRADIENT AT CLOUD HEIGHT.
WIN     CALCULATES THE WIND SPEED IN THE X AND Y DIRECTION AT
        CLOUD HEIGHT.
DIFFUS  COMPUTES THE DIFFUSIVITY AT A SPECIFIED HEIGHT.

CALLED BY RKM
*****
IF(T.LT.TBURST)GO TO 200
IF(Y(6).GT.ZSTAR)GO TO 5
CALL TEMP(Y(6),TA,DTADZ)
GO TO 6
5 TA=TC1+TC2*Y(6)+TC3*Y(6)**2
  DTADZ=TC2+2.*TC3*Y(6)
6 CALL WIN(Y(6),XWCM,YWCM)
  TOP=Y(6)+Y(1)
  CALL WIN(TOP,XWTOP,YWTOP)

```

```

DFQ00010
DFQ00020
DFQ00030
DFQ00040
DFQ00050
DFQ00060
DFQ00070
DFQ00080
DFQ00090
DFQ00100
DFQ00110
DFQ00120
DFQ00130
DFQ00140
DFQ00150
DFQ00160
DFQ00170
DFQ00180
DFQ00190
DFQ00200
DFQ00210
DFQ00220
DFQ00230
DFQ00240
DFQ00250
DFQ00260
DFQ00270
DFQ00280
DFQ00290
DFQ00300
DFQ00310
DFQ00320
DFQ00330
DFQ00340
DFQ00350
DFQ00360
DFQ00370
DFQ00380
DFQ00390
DFQ00400
DFQ00410
DFQ00420
DFQ00430
DFQ00440
DFQ00450
DFQ00460
DFQ00470
DFQ00480
DFQ00490
DFQ00500
DFQ00510
DFQ00520
DFQ00530
DFQ00540
DFQ00550
DFQ00560
DFQ00570
DFQ00580
DFQ00590
DFQ00600
DFQ00610
DFQ00620
DFQ00630
DFQ00640
DFQ00650
DFQ00660
DFQ00670
DFQ00680
DFQ00690
DFQ00700

```


	TDIF=TA-T0	DFQ00710
	TA THE AMBIENT ATMOSPHERIC TEMPERATURE AT CLOUD HEIGHT	DFQ00720
	DTADZ THE TEMPERATURE GRADIENT AT CLOUD HEIGHT	DFQ00730
	XWCM THE WIND SPEED IN THE X DIRECTION AT CLOUD C.O.M.	DFQ00740
	YXCM THE WIND SPEED IN THE Y DIRECTION AT CLOUD C.O.M.	DFQ00750
	XWTOP THE WIND SPEED IN THE X DIRECTION AT THE TOP OF THE CLOUD	DFQ00760
	YWTOP THE WIND SPEED IN THE Y DIRECTION AT THE TOP OF THE CLOUD	DFQ00770
	TR THE RATIO OF CLOUD TEMPERATURE TO AMBIENT TEMPERATURE	DFQ00780
	TR=Y(2)/TA	DFQ00790
	CALCULATE ARVOL, THE SURFACE AREA TO VOLUME RATIO	DFQ00800
	ARVOL=3./Y(1)	DFQ00810
	DEFINITION OF DIFFERENTIAL EQUATIONS	DFQ00820
	YP(1)=ALPHAK*ABS(Y(3))	DFQ00830
	ZZ1=Y(6)	DFQ00840
	KM=DIFFUS(Z0,ZL,ZZ1)	DFQ00850
	GROWTH=KM/Y(1)	DFQ00860
	IF(Y(1).LT.GROWTH)YP(1)=GROWTH	DFQ00870
	YP(2)=-((1.+TR)*ARVOL*Y(2)*YP(1)-Y(3)*(DTADZ)	DFQ00880
	YP(3)=9.8*TR-1.4*ARVOL*Y(3)*YP(1)	DFQ00890
	IF(Y(1)+Y(6).GT.ZINV)YP(3)=0.0	DFQ00900
	YP(4)=XWCM	DFQ00910
	YP(5)=YWCM	DFQ00920
	YP(6)=Y(3)	DFQ00930
	IF(Y(1)+Y(6).GT.ZINV)YP(6)=0.0	DFQ00940
	YP(7)=XWTOP	DFQ00950
	YP(8)=YWTOP	DFQ00960
	GO TO 999	DFQ00970
200	CONTINUE	DFQ00980
	DO 210 I=1,N	DFQ00990
	YP(I)=0.	DFQ01000
210	CONTINUE	DFQ01010
	YP(3)=ACCEL	DFQ01020
	YP(6)=Y(3)	DFQ01030
999	RETURN	DFQ01040
	END	DFQ01050
		DFQ01060
		DFQ01070
		DFQ01080
		DFQ01090
		DFQ01100
		DFQ01110

FUNCTION DIFFUS(Z0,ZL,Z)	DIF00010
COMMON/STARS/USTAR,TSTAR,ZSTAR	DIF00020
COMMON/COEF/AW,CW,BW,DW,AT,CT,BT,DT	DIF00030
*****	DIF00040
PURPOSE	DIF00050
TO CALCULATE THE DIFFUSIVITY AT A GIVEN HEIGHT	DIF00060
INPUTS	DIF00070
Z0 FRICTION HEIGHT IN METERS.	DIF00080
ZL MONIN OBUKHOV LENGTH IN METERS.	DIF00090
Z HEIGHT AT WHICH DIFFUSIVITY IS DESIRED.	DIF00100
CALLED BY ATMCAL, RISE AND DIFEQ	DIF00110
SUBROUTINES AND FUNCTIONS NEEDED	DIF00120
NONE	DIF00130
*****	DIF00140
ZZ=Z	DIF00150
IF(Z.GT.ZSTAR)Z=ZSTAR	DIF00160
NEUTRAL CASE	DIF00170
IF(ABS(ZL).LT.1.E3)GO TO 100	DIF00180
DIFFUS=.4*USTAR*Z	DIF00190
GO TO 999	DIF00200
100 IF(ZL.GT.0.0)GO TO 200	DIF00210
UNSTABLE CASE	DIF00220
S=Z/ZL	DIF00230
IF(S.LT.-2.)GO TO 110	DIF00240
DIFFUS=.4*ABS(ZL*USTAR*S*(1.-16.*S)**(1./4.))	DIF00250
GO TO 999	DIF00260
110 DIFFUS=.4*ABS(ZL*(3./AW)*(-1.*S)**(4./3.))*USTAR	DIF00270
GO TO 999	DIF00280
STABLE CASE	DIF00290
200 S=Z/ZL	DIF00300
S0=Z0/ZL	DIF00310
IF(S.GT.1.5)GO TO 210	DIF00320
DIFFUS=.4*ZL*USTAR*ABS(1./((1./S0+S)+7.))	DIF00330
GO TO 999	DIF00340
210 DIFFUS=.4*ZL*USTAR*ABS(1./BW)	DIF00350
Z=ZZ	DIF00360
999 RETURN	DIF00370
END	DIF00380
	DIF00390
	DIF00400
	DIF00410
	DIF00420
	DIF00430
	DIF00440
	DIF00450
	DIF00460
	DIF00470
	DIF00480
	DIF00490
	DIF00500
	DIF00510
	DIF00520
	DIF00530
	DIF00540

C
C
C

```
FUNCTION DOTPRD(A,B)
DIMENSION A(2),B(2)
DOTPRD IS THE SCALAR PRODUCT OF A AND B
DOTPRD=A(1)*B(1)+A(2)*B(2)
RETURN
END
```

```
DOT00010
DOT00020
DOT00030
DOT00040
DOT00050
DOT00060
DOT00070
DOT00080
```


C
202 STOP
END

DTI00710
DTI00720
DTI00730

```

SUBROUTINE DTERPS(II,X,VAL,NZ)
DIMENSION X(81,4,3),VAL(16),II(4)
*****

PURPOSE
    TO SET UP A ONE DIMENSIONAL ARRAY OF THE VALUES CORRESPONDING
    TO THE CORNERS OF THE CUBE WITHIN A TABULATED ARRAY WITH
    LOWEST CORNER INDICES GIVEN

INPUT
    II    SINGLY DIMENSIONED ARRAY CONTAINING THE INDICES OF THE
          LOWEST CORNER OF THE CUBE
    X     A TRIPLY DIMENSIONED ARRAY CONTAINING THE TABULATED
          VALUES TO BE SET UP. THE FIRST INDEX IS THE COLLAPSED
          INDEX FOR THE FIRST TWO INDICES OF A FOUR-DIMENSIONAL
          ARRAY
    NZ    THE RANGE OF THE FIRST INDEX OF THE FOUR-DIMENSIONAL
          ARRAY

OUTPUT
    VAL   SINGLY DIMENSIONED ARRAY CONTAINING THE VALUES OF X
          FOR THE 16 CORNER POINTS OF THE CUBE

CALLED BY MOMENT
*****
M=0
DO 104 L=1,2
  LX=L + II(4) - 1
  DO 103 K=1,2
    KX=K + II(3) - 1
    DO 102 JI=1,2
      JIX=(JI + II(2) - 2)*NZ
      DO 101 IJ=1,2
        IJX=JIX + IJ + II(1) - 1
        M=M+1
        VAL(M)=X(IJX,KX,LX)
      CONTINUE
    CONTINUE
  CONTINUE
CONTINUE
RETURN
END

```

```

DTS00010
DTS00020
DTS00030
DTS00040
DTS00050
DTS00060
DTS00070
DTS00080
DTS00090
DTS00100
DTS00110
DTS00120
DTS00130
DTS00140
DTS00150
DTS00160
DTS00170
DTS00180
DTS00190
DTS00200
DTS00210
DTS00220
DTS00230
DTS00240
DTS00250
DTS00260
DTS00270
DTS00280
DTS00290
DTS00300
DTS00310
DTS00320
DTS00330
DTS00340
DTS00350
DTS00360
DTS00370
DTS00380
DTS00390
DTS00400
DTS00410
DTS00420
DTS00430
DTS00440
DTS00450
DTS00460
DTS00470
DTS00480

```

```

SUBROUTINE DUSTCL(NEWATM,NATHOS,ZTMP,TMPMES,ZWUND,WNDMES,PHI,
1 THUND,NEWSRC,CHWT,NCHRG,DETDEP,NSOIL,DSOD,
2 LOSTRN,TRNCOR,RECCOR,EDGE,OBSCOR,SPCHT,NEWTIM,
3 TIME,TRNLOS,CNTRD,HEIGHT,CENWTH,SPCWTH,NCPTS,CPTS,NERR,
4 NCHS,SRCBAS,SIDE1,SIDE2,DHDT)
LOGICAL NEWATM,NEWSRC,LOSTRN,EDGE,NEWTIM,HORIZ,ERR
LOGICAL SWITCH,CHANGE,DHDT
DIMENSION ZTMP(2),TMPMES(2),ZWUND(2),WNDMES(2),TRNCOR(3)
1 ,RECCOR(3),CPTS(2,6),ORIG(2),TRNFRM(2,2),TRN(3),REC(3)
2 ,CNTRD(2),OBSCOR(2),DIR(2)
3 ,SRCBAS(2),SIDE1(2),SIDE2(2),TEMP(2),NCHS(2)
REAL KZ,KX
COMMON /GEOM/COSTH2,SINTH,SINTH2,VISEXT,RTPI,SCRN(2)
COMMON /MODE/ HORIZ
COMMON /WINDPRM/ DXZO,DYXO,DZO,UO,UM,DN,ZINV
COMMON /CLOCK/ FTIME,TWIND
COMMON /MOS/DIFF(2,200),NCHTOT,PRSEP(200),NTOT,NARY,ITOT,
+ COOR(2,200),TSTAG(200),DMMY(401)
COMMON /ARRAY/OVLAP,AREA,PERIM,PRJARY,CENDIF
COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
COMMON /CARB/RCARB1,RCARB2
COMMON /ACL/CWINDS,CWINDC,CWINDW
COMMON /TRAN/VTR,KZ,KX,TTR,XTR,ZTR,QPUFF(3),SWITCH,CHANGE
COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK
DATA ONEM/-1./
*****

PURPOSE
DUSTCL CALCULATES DUST CLOUD DIMENSIONS AND TRANSMITTANCES
THROUGH DUST CLOUDS FOR GIVEN METEOROLOGICAL DATA, SOIL TYPE,
EXPLOSIVE CHARACTERISTICS, AND WAVELENGTH.

SEE COMMENTS IN DRTRAN FOR DETAILS.

SUBROUTINES CALLED
ATMCAL ACCEPTS METEOROLOGICAL DATA AS ARGUMENTS AND COMPUTES
NECESSARY PARAMETERS IN COMMON /WINDPRM/, /EKWIND/,
/EKTEMP/, /STARS/

SOURCE ACCEPTS SOIL, CHARGE, AND WAVELENGTH SPECIFICATIONS
AS INPUT AND COMPUTES NECESSARY PARAMETERS AND INITIAL
VALUES IN COMMON /PRTINF/, /BUOYCL/ AND /CARB/

SETUP ACCEPTS THE USER DEFINED COORDINATES OF THE CHARGES
AND CONVERTS THEM TO THE INTERNAL (LOCAL) COORDINATE
SYSTEM. ALSO COMPUTES THE AREA AND PERIMETER OF THE
BOUNDING PARALLELOGRAM AND OVERLAP DISTANCE OF THE
CHARGES AND RETURNS THEM IN COMMON /ARRAY/ AND
/SEPRTN/.

RISE GIVEN CLOUD DIMENSIONS DURING BUOYANT RISE DEVELOPMENT
OF CLOUD, RISE CALCULATES THE DIMENSIONS AT A LATER
TIME

CLDIM DETERMINES THE OUTPUT VARIABLES DESCRIBING THE CLOUD
DIMENSIONS.

TRNCAL CONTROLLING ROUTINE FO THE CALCULATION OF TRANSMITTANCES.
*****
IF(LOSTRN.OR.EDGE)GO TO 101
NERR=4
GO TO 999
101 IF(.NOT.NEWATM) GO TO 200
THETAX=THUND*PIRAD
CALL ATMCAL(NATHOS,ZTMP,TMPMES,ZWUND,WNDMES,PHI,THETAX,DHDT,ERR)
99999 IF(.NOT.ERR)GO TO 155

```

```

NERR=7
GO TO 999
155 CONTINUE

CCCC
COMPUTE THE ROTATION TRANSFORMATION MATRIX TO CONVERT USER
DEFINED COORDINATES INTO LOCAL COORDINATES WITH X AXIS IN
THE WIND DIRECTION.

THETAX=THWIND*PIRAD
TRNFRM(1,1)=COS(THETAX)
TRNFRM(2,2)=TRNFRM(1,1)
TRNFRM(1,2)=SIN(THETAX)
TRNFRM(2,1)=-TRNFRM(1,2)
200 CONTINUE
IF(.NOT. NEWSRC) GO TO 300
TWIND=1.E5
TTR=1.E5
TPRES=0.
DEL=.001
DO 250 I=1,2
IF(MARY.GT.1)SRCBAS(I)=COOR(I,1)
ORIG(I)=SRCBAS(I)
250 CONTINUE
CALL SOURCE(CHWT,NCHRG,DETDEP,NSOIL,DSOD)
CALL SETUP(NCHS,SRCBAS,SIDE1,SIDE2,TRNFRM)
300 CONTINUE
IF(.NOT. LOSTRN) GO TO 400

CCCC
CONVERT TRNCOR AND RECCOR TO LOCAL COORDINATES WITH ORIGIN AT
SRCBAS AND X AXIS IN WIND DIRECTION.

TRN(3)=TRNCOR(3)
REC(3)=RECCOR(3)
DO 320 I=1,2
TRN(I)=0.
REC(I)=0.
DO 310 J=1,2
TRN(I)=TRN(I)+TRNFRM(I,J)*(TRNCOR(J)-ORIG(J))
REC(I)=REC(I)+TRNFRM(I,J)*(RECCOR(J)-ORIG(J))
310 CONTINUE
320 CONTINUE
400 CONTINUE
IF(.NOT. EDGE) GO TO 500

CCCC
COMPUTE A UNIT VECTOR IN THE DIRECTION OF THE OBSERVERS LINE
OF SIGHT AND A UNIT VECTOR PERPENDICULAR TO THE LINE OF SIGHT

CALL VSUM(ORIG,OBSCOR,ONEM,DIR)
CALL UNIT(DIR,DIR,RANGE)
COSTH=0.
SINTH=0.
DO 410 J=1,2
COSTH=COSTH+TRNFRM(1,J)*DIR(J)
SINTH=SINTH+TRNFRM(2,J)*DIR(J)
410 CONTINUE
SINTH2=SINTH*SINTH
COSTH2=COSTH**2
SCRN(1)=SINTH
SCRN(2)=-COSTH

CCCC
COMPUTE THE PROJECTION OF EACH DIFFERENCE VECTOR DIFF ONTO THE
VECTOR PERPENDICULAR TO THE LINE OF SIGHT, (DIFF(1,J),DIFF(2,J))
IS THE VECTOR FROM THE REFERENCE CHARGE TO THE JTH CHARGE
LOCATION IN THE INTERNAL COORDINATE SYSTEM.

PARY1=0.0
PARY2=0.0
DO 420 J=1,NCHTOT
DO 415 I=1,2
TEMP(I)=DIFF(I,J)

```

```

DUS00710
DUS00720
DUS00730
DUS00740
DUS00750
DUS00760
DUS00770
DUS00780
DUS00790
DUS00800
DUS00810
DUS00820
DUS00830
DUS00840
DUS00850
DUS00860
DUS00870
DUS00880
DUS00890
DUS00900
DUS00910
DUS00920
DUS00930
DUS00940
DUS00950
DUS00960
DUS00970
DUS00980
DUS00990
DUS01000
DUS01010
DUS01020
DUS01030
DUS01040
DUS01050
DUS01060
DUS01070
DUS01080
DUS01090
DUS01100
DUS01110
DUS01120
DUS01130
DUS01140
DUS01150
DUS01160
DUS01170
DUS01180
DUS01190
DUS01200
DUS01210
DUS01220
DUS01230
DUS01240
DUS01250
DUS01260
DUS01270
DUS01280
DUS01290
DUS01300
DUS01310
DUS01320
DUS01330
DUS01340
DUS01350
DUS01360
DUS01370
DUS01380
DUS01390
DUS01400

```


415	CONTINUE	DUS01410
	PRSEP(J)=DOTPRD(TEMP,SCRN)	DUS01420
	X=PRSEP(J)	DUS01430
	IF(X.LT.0.0)GO TO 416	DUS01440
	IF(X.GT.PARY1)PARY1=X	DUS01450
	GO TO 420	DUS01460
416	IF(X.LT.PARY2)PARY2=X	DUS01470
420	CONTINUE	DUS01480
	PRJARY=(PARY1-PARY2)/2.	DUS01490
	CENDIF=(PARY1+PARY2)/2.	DUS01500
500	CONTINUE	DUS01510
	IF(NARY.EQ.3)GO TO 600	DUS01520
	IF(NEWTIM) CALL RISE(TPRES,TIME,DEL)	DUS01530
600	IF(.NOT.EDGE) GO TO 650	DUS01540
	FTIME=TIME	DUS01550
	CALL CLDIM(ONTRO,HEIGHT,CENWTH,SPCWT,HCPTS,CPTS,ERR)	DUS01560
	IF(.NOT.ERR)GO TO 650	DUS01570
	NERR=6	DUS01580
	GO TO 999	DUS01590
650	CONTINUE	DUS01600
	IF(.NOT.LOSTRN)GO TO 999	DUS01610
C	CALL TRNCAL(TRN,REC,TIME,TRNLOS)	DUS01620
999	RETURN	DUS01630
	END	DUS01640
		DUS01650

```

FUNCTION ERF(X)
CALCULATES THE ERROR FUNCTION

INPUT
  X    VALUE AT WHICH ERROR FUNCTION IS DESIRED
FUNCTIONS AND SUBROUTINES NEEDED
  NONE
*****
  DIMENSION P(3,5),Q(3,5)
  DATA RPI/.5641896/
  DATA P/2.138533E+01,7.373888E+00,-4.257996E-02,
+       1.722276E+00,6.865018E+00,-1.960690E-01,
+       3.166529E-01,3.031799E+00,-5.168823E-02,
+       0.,5.631696E-01,0.,
+       0.,4.318779E-05,0.,
  DATA Q/1.895226E+01,7.373961E+00,1.509421E-01,
+       7.843746E+00,1.518491E+01,9.214524E-01,
+       1.000000E+00,1.279553E+01,1.000000E+00,
+       0.,5.354217E+00,0.,
+       0.,1.000000E+00,0.,
  AX=ABS(X)
  ERFC=0.0
  IF(AX.GT.11.0) GO TO 300
  X2=AX*AX
  I=2
  IF(AX.LT.0.5) I=1
  IF(AX.GT.4.) I=3
  IF(I-2) 10,20,30
10  N=3
   Z=X2
   GO TO 40
20  N=5
   Z=AX
   GO TO 40
30  N=3
   Z=1./X2
40  SP=P(I,N)
   SQ=Q(I,N)
   N1=N-1
   DO 50 K=1,N1
     J=N-K
     SP=SP*Z+P(I,J)
     SQ=SQ*Z+Q(I,J)
     IF(I-2) 60,70,80
60  ERFC=1.0-X*SP/SQ
   ERF=1.-ERFC
   RETURN
70  ERFC=EXP(-X2)*SP/SQ
   GO TO 300
80  ERFC=EXP(-X2)/AX*(RPI+SP/(SQ*X2))
300 IF(X.LT.0.0) ERFC=2.0-ERFC
   ERF=1.-ERFC
   RETURN
END

```

```

ERF 00120
ERF 00010
ERF 00020
ERF 00030
ERF 00040
ERF 00050
ERF 00060
ERF 00070
ERF 00080
ERF 00090
ERF 00100
ERF 00110
ERF 00130
ERF 00140
ERF 00150
ERF 00160
ERF 00170
ERF 00180
ERF 00190
ERF 00200
ERF 00210
ERF 00220
ERF 00230
ERF 00240
ERF 00250
ERF 00260
ERF 00270
ERF 00280
ERF 00290
ERF 00300
ERF 00310
ERF 00320
ERF 00330
ERF 00340
ERF 00350
ERF 00360
ERF 00370
ERF 00380
ERF 00390
ERF 00400
ERF 00410
ERF 00420
ERF 00430
ERF 00440
ERF 00450
ERF 00460
ERF 00470
ERF 00480
ERF 00490
ERF 00500
ERF 00510
ERF 00520
ERF 00530
ERF 00540
ERF 00550
ERF 00560
ERF 00570
ERF 00580

```

<pre> SUBROUTINE FIT(X,F,A,B,C) QUADRATIC FIT TO THREE POINTS USING NEWTON'S FUNDAMENTAL FORMULA INPUTS X - 3 VALUES OF THE INDEPENDENT VARIABLE F - 3 FUNCTION VALUES CORRESPONDING TO THE X VALUES OUTPUTS A - COEFFICIENT OF THE X**2 TERM B - COEFFICIENT OF THE X TERM C - CONSTANT TERM C***** DIMENSION X(3),F(3) H=X(2)-X(1) DF1=(F(2)-F(1))/H DF2=(F(3)-2.*F(2)+F(1))/(2.*H**2) A=DF2 B=DF1-DF2*(X(2)+X(1)) C=F(1)+X(1)*(X(2)*DF2-DF1) RETURN END </pre>	<pre> FIT00190 FIT00010 FIT00020 FIT00030 FIT00040 FIT00050 FIT00060 FIT00070 FIT00080 FIT00090 FIT00100 FIT00110 FIT00120 FIT00130 FIT00140 FIT00150 FIT00160 FIT00170 FIT00180 FIT00200 FIT00210 FIT00220 FIT00230 FIT00240 FIT00250 FIT00260 FIT00270 FIT00280 </pre>
--	--

```

FUNCTION FUNCT(X,Z)
LOGICAL HORIZ,SKIP
COMMON /CLOCK/ TIME,TWIND
COMMON/MODE/HORIZ
COMMON/SKIPIT/SKIP

```

PURPOSE

TO SUPPLY A TRANSMITTANCE FUNCTION FOR THE CONTOUR TRACING
ROUTINE IN ORDER TO DETERMINE THE CLOUD EDGE.

INPUT

X THE HORIZONTAL COORDINATE IN METERS
Z THE VERTICAL COORDINATE IN METERS

OUTPUT

RETURNS THE LOG OF THE OPTICALLY WEIGHTED CL VALUE (AT
VISIBLE WAVELENGTHS) FOR THE LINE OF SIGHT SPECIFIED BY X,Z

FUNCTIONS CALLED

CWIND

CALLED BY GFUN, CLIMB, GRAD2

```

HORIZ=.TRUE.
SKIP=.TRUE.
Y=0.
EXT1=0.0
EXT2=0.0
IF(Z.LE.0.)GO TO 100
EXT1=CWIND(X,Y,Z,TIME)
IF(TIME.LE.TWIND)GO TO 10
EXT2=CSPHER(X,Y,Z,TIME)
10 EXT=EXT1+EXT2
IF(EXT.LE.1.E-30)GO TO 100
FUNCT=ALOG(EXT)
GO TO 999
100 FUNCT=-30.
999 CONTINUE
RETURN
END

```

```

FUC00010
FUC00020
FUC00030
FUC00040
FUC00050
FUC00060
FUC00070
FUC00080
FUC00090
FUC00100
FUC00110
FUC00120
FUC00130
FUC00140
FUC00150
FUC00160
FUC00170
FUC00180
FUC00190
FUC00200
FUC00210
FUC00220
FUC00230
FUC00240
FUC00250
FUC00260
FUC00270
FUC00280
FUC00290
FUC00300
FUC00310
FUC00320
FUC00330
FUC00340
FUC00350
FUC00360
FUC00370
FUC00380
FUC00390
FUC00400
FUC00410
FUC00420
FUC00430
FUC00440
FUC00450
FUC00460
FUC00470

```

<pre> SUBROUTINE GAMMA(XX,GX,IER) SUBROUTINE GAMMA PURPOSE COMPUTES THE GAMMA FUNCTION FOR A GIVEN ARGUMENT USAGE CALL GAMMA(XX,GX,IER) DESCRIPTION OF PARAMETERS XX -THE ARGUMENT FOR THE GAMMA FUNCTION GX -THE RESULTANT GAMMA FUNCTION IER -THE RESULTANT ERROR CODE WHERE IER=0 NO ERROR IER=1 XX IS WITHIN .000001 OF BEING A NEGATIVE INTEGER IER=2 XX GT 57, OVERFLOW, GX SET TO 1.E32 COMMENTS NONE SUBROUTINES AND FUNCTIONS NONE METHOD THE RECURSION RELATION AND POLYNOMIAL APPROXIMATION BY C. HASTINGS, JR., 'APPROXIMATIONS FOR DIGITAL COMPUTERS', PRINCETON UNIVERSITY PRESS, 1955 4 IF(XX-57.) 6,6,4 IER=2 GX=1.E32 RETURN 6 X=XX ERR=1.0E-6 IER=0 CX=1.0 IF(X-2.0) 50,50,15 10 IF(X-2.0) 110,110,15 15 X=X-1.0 CX=GX*X GO TO 10 50 IF(X-1.0) 60,120,110 SEE IF X IS NEAR NEGATIVE INTEGER OR ZERO 60 IF(X-ERR) 62,62,80 62 Y=FLOAT(INT(X))-X IF(ABS(Y)-ERR) 130,130,70 X NOT NEAR A NEGATIVE INTEGER OR ZERO 70 IF(X-1.0) 80,80,110 80 CX=GX/X X=X+1.0 GO TO 70 110 Y=X-1.0 CY=1.0+Y*(-0.5771017+Y*(0.9858540+Y*(-0.8764218+Y*(0.8328212+ Y*(-0.5684729+Y*(0.2548205+Y*(-0.05149930)))))) CX=GX*CY 120 RETURN 130 IER=1 RETURN END </pre>	<pre> GAM00310 GAM00010 GAM00020 GAM00030 GAM00040 GAM00050 GAM00060 GAM00070 GAM00080 GAM00090 GAM00100 GAM00110 GAM00120 GAM00130 GAM00140 GAM00150 GAM00160 GAM00170 GAM00180 GAM00190 GAM00200 GAM00210 GAM00220 GAM00230 GAM00240 GAM00250 GAM00260 GAM00270 GAM00280 GAM00290 GAM00300 GAM00320 GAM00330 GAM00340 GAM00350 GAM00360 GAM00370 GAM00380 GAM00390 GAM00400 GAM00410 GAM00420 GAM00430 GAM00440 GAM00450 GAM00460 GAM00470 GAM00480 GAM00490 GAM00500 GAM00510 GAM00520 GAM00530 GAM00540 GAM00550 GAM00560 GAM00570 GAM00580 GAM00590 GAM00600 GAM00610 GAM00620 GAM00630 GAM00640 GAM00650 GAM00660 </pre>
---	--

C
C
C

FUNCTION GFUN(S)

GFUN IS THE RESTRICTION OF THE TWO DIMENSIONAL FUNCTION, F, TO
A LINE. I.E. FORM $G(S)=F(X,Y)$, WHERE $(X,Y)=BASE+S*DIR$.

EXTERNAL FUNCT

DIMENSION

P(2)

COMMON/LINE/BASE(2),DIR(2),DFDS/SPECS/RES,DELTA,THETAN,CON

CALL VSUM(BASE,DIR,S,P)

GFUN=FUNCT(P(1),P(2))

RETURN

END

GFU00010

GFU00020

GFU00030

GFU00040

GFU00050

GFU00060

GFU00070

GFU00080

GFU00090

GFU00100

GFU00110

GFU00120


```

SUBROUTINE GRAND(U,TR,XNORM,TIME,TIVEH,VDIR,VALUE)
ROUTINE TO EVALUTE THE INTEGRAND FOR THE TRAPEZOIDAL INTEGRATION
FOR FINDING THE OPTICALLY WEIGHTED CONCENTRATION ALONG THE LINE OF
SIGHT WHERE THE DUST IS GENERATED BY A VEHICLE.
INPUTS
  U      - A UNIT VECTOR ALONG THE LINE OF SIGHT
  TR     - THE COORDINATES OF THE TRANSMITTER IN THE LOCAL COORDINATE
          SYSTEM
  XNORM  - DISTANCE BETWEEN THE TRANSMITTER AND RECEIVER
  TIME   - PRESENT TIME AT WHICH A TRANSMITTANCE IS WANTED
  TIVEN  - TIME THAT THE VEHICLE HAS TRAVELED
  VDIR   - VECTOR CONTAINING DESCRIBING THE VEHICLE DIRECTION AND
          SPEED
OUTPUT
  VALUE  - VALUE OF THE INTEGRAND
FUNCTIONS AND SUBROUTINES NEEDED
  CONLEN - TO FIND THE LENGTH OF THE INTERSECTION OF THE LINE OF
          SIGHT AND THE TILTED CYLINDER
C*****
  DIMENSION U(3),TR(3),VDIR(2),VP(2)
  COMMON/MO5/DMMY(604),DMM(600),
  +      ICOUNT,TIMES(25),XC0(3,25),XC1(3,25),RT(3,25),
  +      RB(3,25),Z2(3,25)
  COMMON /IQUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLINT,KSTOR,NPLOTU
  COMMON/VL/VLOAD
  COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDECK
C
C  FIND THE VEHICLE POSITION AT TIME TIVEH
C
  VP(1)=TIVEH*VDIR(1)
  VP(2)=TIVEH*VDIR(2)
  TOF=TIME-TIVEH
  DO 10 I=1,ICOUNT
    IND=I
    IF(TOF.LT.TIMES(I))GO TO 20
  10 CONTINUE
C
C  IF TOF (TIME OF FLIGHT) IS GREATER THAN TABULATED VALUES IT IS
C  ASSUMED THE THE CLOUD HAS DISSIPATED
C
  XC=X1+Z*X0  IS THE LINE THRUH THE CENTER OF THE CYLINDER
C
  GO TO 50
  20 X0=TOF*(XC0(1,IND)*TOF+XC0(2,IND))+XC0(3,IND)
  X1=TOF*(XC1(1,IND)*TOF+XC1(2,IND))+XC1(3,IND)
  RAD=TOF*(RT(1,IND)*TOF+RT(2,IND))+RT(3,IND)
  HTTOP=TOF*(Z2(1,IND)*TOF+Z2(2,IND))+Z2(3,IND)
  HTBOT=0.0
  RTOP=RAD
  RBOT=RAD
  XCEN=VP(1)+(X1*HTTOP+X0)
  YCEN=VP(2)
  XB=VP(1)+(X1*HTBOT+X0)
  YB=VP(2)
  IF(U(3).LT.1.E-06)GO TO 30
C
C  COMPUTE INTERSECTION LENGTH FOR NON HORIZONTAL LINES OF SIGHT
C

```

GRD00320
 GRD00010
 GRD00020
 GRD00030
 GRD00040
 GRD00050
 GRD00060
 GRD00070
 GRD00080
 GRD00090
 GRD00100
 GRD00110
 GRD00120
 GRD00130
 GRD00140
 GRD00150
 GRD00160
 GRD00170
 GRD00180
 GRD00190
 GRD00200
 GRD00210
 GRD00220
 GRD00230
 GRD00240
 GRD00250
 GRD00260
 GRD00270
 GRD00280
 GRD00290
 GRD00300
 GRD00310
 GRD00330
 GRD00340
 GRD00350
 GRD00360
 GRD00370
 GRD00380
 GRD00390
 GRD00400
 GRD00410
 GRD00420
 GRD00430
 GRD00440
 GRD00450
 GRD00460
 GRD00470
 GRD00480
 GRD00490
 GRD00500
 GRD00510
 GRD00520
 GRD00530
 GRD00540
 GRD00550
 GRD00560
 GRD00570
 GRD00580
 GRD00590
 GRD00600
 GRD00610
 GRD00620
 GRD00630
 GRD00640
 GRD00650
 GRD00660
 GRD00670
 GRD00680
 GRD00690
 GRD00700

	CALL CONLEN(U,TR,HTTOP,HTBOT,XCEN,YCEN,RTOP,RBOT,XB,YB,XNORM,PLEN)	GRD00710
	GO TO 40	GRD00720
C		GRD00730
C	DETERMINE LENGTH OF INTERSECTION FOR A HORIZONTAL LINE OF SIGHT	GRD00740
	30 IF(HTTOP,LT,TR(3))GO TO 35	GRD00750
	A=U(1)**2+U(2)**2	GRD00760
	B=U(1)*(TR(1)-XCEN)+U(2)*(TR(2)-YCEN)	GRD00770
	C=(TR(1)-XCEN)**2+(TR(2)-YCEN)**2-RAD**2	GRD00780
	X=B**2-A*C	GRD00790
	IF(X,LT,0.0)GO TO 35	GRD00800
	P1=(-B+SQRT(X))/A	GRD00810
	P2=(-B-SQRT(X))/A	GRD00820
	IF(P1,GT,XNORM,AND,P2,GT,XNORM)GO TO 35	GRD00830
	IF(P2,LT,0.0,AND,P1,LT,0.0)GO TO 35	GRD00840
	PLEN=AMIN1(P1,XNORM)-AMAX1(P2,0.0)	GRD00850
	GO TO 40	GRD00860
35	PLEN=0.0	GRD00870
40	VOL=PI*HTTOP*(RAD**2)	GRD00880
	VALUE=VLOAD*PLEN/VOL	GRD00890
	GO TO 999	GRD00900
50	VALUE=0.0	GRD00910
999	RETURN	GRD00920
	END	GRD00930
		GRD00940

```

C SUBROUTINE GREEN(Z,Z1,T,ALPHA,T0,IER)
C *****
C
C PURPOSE
C   TO COMPUTE THE GENERALIZED GREENS FUNCTION
C   USES GREEN1
C   SEE GREEN1 FOR ARGUMENT LIST
C *****
C REAL N,M
C COMMON /WNDPRM/DXZ0,DYX0,DZ0,U0,M,N,ZINV
C
C IF(N.EQ. 1.) GO TO 2
C X2=2,-N
C AT=ALPHA*T
C T0=0.
C IF(AT.GE. Z1) RETURN
C CALL GREEN1((Z+AT)**X2,Z1**X2,X2*X2*T,(N-1.)/X2,T1,IER)
C T1=T1*X2*Z1**(<1.-N)
C U=1.
C T2=0.
C IF(ABS(ALPHA).LT. 1.E-4) GO TO 1
C ZM2=Z-Z1+AT
C X2=N+1
C AN1=ALPHA*X2
C ZMZN=Z1**X2 - (Z1-AT)**X2
C ARG=(-AN1*ZM2*ZM2)/(4.*ZMZN)
C IF(ARG.LT. -70.) GO TO 3
C T2=SQRT(AN1/(4.*3.1415926*ZMZN))*EXP(ARG)
C 3 IF(T1.LT.1.E-30 .AND. T2.LT.1.E-30) RETURN
C
C CALCULATION OF MIXING RATIO, U, BY N=1 ANALOGY
C CALL GREEN1(Z+AT,Z1,T,0.,T1U,IER)
C X2=2.
C AN1=ALPHA*X2
C ZMZN=Z1**X2 - (Z1-AT)**X2
C T2U=0.
C ARG=(-AN1*ZM2*ZM2)/(4.*ZMZN)
C IF(ARG.LT. -70.) GO TO 4
C T2U=SQRT(AN1/(4.*3.1415926*ZMZN))*EXP(ARG)
C 4 IF(T1U.LT.1.E-30 .AND. T2U.LT.1.E-30) GO TO 1
C CALL GREEN1(Z,Z1,T,ALPHA,G,IER)
C U=(G-T2U)/(T1U-T2U)
C 1 IF(U.LT. 0.) U=0.
C IF(U.GT. 1.) U=1.
C
C COMBINE LIMITING SOLUTIONS WITH DETERMINED MIXING RATIO
C
C T0=U*T1 + (1.-U)*T2
C RETURN
C 2 CALL GREEN1(Z,Z1,T,ALPHA,T0,IER)
C RETURN
C END

```

```

GRE00010
GRE00020
GRE00030
GRE00040
GRE00050
GRE00060
GRE00070
GRE00080
GRE00090
GRE00100
GRE00110
GRE00120
GRE00130
GRE00140
GRE00150
GRE00160
GRE00170
GRE00180
GRE00190
GRE00200
GRE00210
GRE00220
GRE00230
GRE00240
GRE00250
GRE00260
GRE00270
GRE00280
GRE00290
GRE00300
GRE00310
GRE00320
GRE00330
GRE00340
GRE00350
GRE00360
GRE00370
GRE00380
GRE00390
GRE00400
GRE00410
GRE00420
GRE00430
GRE00440
GRE00450
GRE00460
GRE00470
GRE00480
GRE00490
GRE00500
GRE00510
GRE00520
GRE00530
GRE00540
GRE00550
GRE00560
GRE00570
GRE00580
GRE00590

```

SUBROUTINE GREEN1(Z,Z1,T,NU,BI,IER)

SUBROUTINE GREEN1

PURPOSE

COMPUTE THE I BESSEL FUNCTION FOR A GIVEN ARGUMENT AND ORDER
AND MULTIPLY BY AN APPROPRIATE POWER OF THE ARGUMENT
AND AN EXPONENTIAL IN ORDER TO CALCULATE THE GREENS
FUNCTION FOR THE WIND DIFFUSION EQUATION

USAGE

CALL GREEN1(Z,Z1,T,NU,BI,IER)

DESCRIPTION OF PARAMETERS

Z,Z1,T -THE ARGUMENTS OF THE FUNCTION DESIRED

NU -THE ORDER OF THE I BESSEL FUNCTION

BI -THE RESULTANT BESSEL FUNCTION

IER -RESULTANT ERROR CODE WHERE

IER=-1 EXPONENTIAL UNDERFLOW (NON-FATAL), BI SET TO 0.0

IER=0 NO ERROR

IER=1 NU NEAR NEGATIVE INTEGER

IER=2 OVERFLOW IN GAMMA

IER=3 UNDERFLOW, BI .LT. 1.E-32, BI SET TO 0.0

IER=4 OVERFLOW, X .GT. 90 WHERE X .GT. N

IER=5 X IS NEGATIVE

REMARKS

NU IS A REAL NUMBER

N AND X MUST BE .GE. ZERO

THIS SUBROUTINE IS A MODIFICATION OF BESI WHICH COMPUTES THE

I BESSEL FUNCTION FOR INTEGER ORDERS. THE CHANGE REQUIRES

USE OF THE GAMMA FUNCTION FOR COMPUTING THE FIRST TERM OF THE

SERIES. THE SUCCESSIVE TERMS ARE CALCULATED WITH THE SAME

RECURSION FORMULA AND THE ASYMPTOTIC APPROXIMATION IS ALSO

UNCHANGED. BESI IS IN THE IBM SYSTEM/360 SCIENTIFIC

SUBROUTINE PACKAGE. MODIFICATIONS MADE BY D. DVORE, AERODYNE

RESEARCH INC. JANUARY 15, 1979.

SUBROUTINES AND FUNCTIONS REQUIRED

GAMMA WHICH COMPUTES THE GAMMA FUNCTION

METHOD

COMPUTES I BESSEL FUNCTION USING SERIES OR ASYMPTOTIC
APPROXIMATION DEPENDING ON THE RANGE OF THE ARGUMENT.

CALLED BY MOMENT

REAL NU

X=2.*SQRT(Z*Z1)/T

CHECK FOR ERRORS IN NU AND X AND EXIT IF ANY ARE PRESENT

IER=0

BI=1.0

IF(NU)10,15,10

10 IF(X)160,20,20

15 IF(X)160,17,20

17 ARG=-(Z+Z1)/T

IF(ARG .LT. -80.) GO TO 170

BI=EXP(ARG)/T

RETURN

DEFINE TOLERANCE

20 TOL=1 E-3

IF ARGUMENT GT 12 AND GT NU, USE ASYMPTOTIC FORM

GRV00490
GRV00010
GRV00020
GRV00030
GRV00040
GRV00050
GRV00060
GRV00070
GRV00080
GRV00090
GRV00100
GRV00110
GRV00120
GRV00130
GRV00140
GRV00150
GRV00160
GRV00170
GRV00180
GRV00190
GRV00200
GRV00210
GRV00220
GRV00230
GRV00240
GRV00250
GRV00260
GRV00270
GRV00280
GRV00290
GRV00300
GRV00310
GRV00320
GRV00330
GRV00340
GRV00350
GRV00360
GRV00370
GRV00380
GRV00390
GRV00400
GRV00410
GRV00420
GRV00430
GRV00440
GRV00450
GRV00460
GRV00470
GRV00480
GRV00500
GRV00510
GRV00520
GRV00530
GRV00540
GRV00550
GRV00560
GRV00570
GRV00580
GRV00590
GRV00600
GRV00610
GRV00620
GRV00630
GRV00640
GRV00650
GRV00660
GRV00670
GRV00680
GRV00690
GRV00700

	IF(X-12.)40,40,30	GRV00710
30	IF(X-ABS(NU))40,40,110	GRV00720
CCC	COMPUTE FIRST TERM OF SERIES AND SET INITIAL VALUE OF THE SUM	GRV00730
40	XX=X/2.	GRV00740
	N=INT(NU)	GRV00750
	FN=N	GRV00760
	R=NU-FN	GRV00770
	CALL GAMMA(1,+NU,GR,IER)	GRV00780
	IF(IER.EQ. 0) GO TO 60	GRV00790
50	BI=0.0	GRV00800
	RETURN	GRV00810
60	TERM=1./GR	GRV00820
70	BI=TERM	GRV00830
	XX=XX*XX	GRV00840
CCC	COMPUTE TERMS, STOPPING WHEN ABS(TERM) LE ABS(SUM OF TERMS)*TOLERA	GRV00850
	DO 90 K=1,1000	GRV00860
	IF(ABS(TERM)-ABS(BI*TOL))95,95,80	GRV00870
80	FK=K	GRV00880
	FK=FK*(NU+FK)	GRV00890
	TERM=TERM*(XX/FK)	GRV00900
90	BI=BI+TERM	GRV00910
95	ARG=-(Z+Z1)/T	GRV00920
	IF(ARG.LT. -80.) GO TO 170	GRV00930
	BI=BI*(Z1/T)**NU*EXP(ARG)/T	GRV00940
CCC	RETURN BI AS ANSWER	GRV00950
100	RETURN	GRV01000
CCC	X GT 12 AND X GT NU, SO USE ASYMPTOTIC APPROXIMATION	GRV01010
110	FN=4.*NU*NU	GRV01020
115	XX=1./(8.*X)	GRV01030
	TERM=1.	GRV01040
	BI=1.	GRV01050
	DO 130 K=1,30	GRV01060
	IF(ABS(TERM)-ABS(BI*TOL))140,140,120	GRV01070
120	FK=(2*K-1)**2	GRV01080
	TERM=TERM*XX*(FK-FN)/FLOAT(K)	GRV01090
130	BI=BI+TERM	GRV01100
CCC	SIGNIFICANCE LOST AFTER 30 TERMS, TRY SERIES	GRV01110
	GO TO 40	GRV01120
140	PI=3.141592653	GRV01130
	ARG=X-(Z+Z1)/T	GRV01140
	IF(ARG.LT. -80.) GO TO 170	GRV01150
	BI=BI*(Z1/Z)**(NU/2.)*EXP(ARG)/SQRT(2.*PI*X)/T	GRV01160
	GO TO 100	GRV01170
160	IER=5	GRV01180
	GO TO 100	GRV01190
170	BI=0.0	GRV01200
	GO TO 50	GRV01210
	END	GRV01220
		GRV01230
		GRV01240
		GRV01250
		GRV01260
		GRV01270
		GRV01280

```

SUBROUTINE MOMENT(VGRAV,ZIN,H,TIN,Q,XBAR,SIGW2,SIGP2)
REAL M,N,NM
DIMENSION AL(9),Z(9),T(9),XB(81,4,3),SW(81,4,3),SP(81,4,3),NM(9)
DIMENSION VAL(16),XVAL(8),W(8),XI(4),IB(4),NTC(4),II(4),X(9,4)
LOGICAL FIRST
COMMON /WINDPRM/DXZ0,DYX0,DZ0,U0,M,N,ZINV
COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
EQUIVALENCE (Z(1),X(1,1)),(T(1),X(1,2)),(AL(1),X(1,3))
EQUIVALENCE (NM(1),X(1,4))
DATA FIRST/.TRUE./,IB/4,4,1,2/,ITC/11/,HREF/1./
*****

PURPOSE
    TO CONVERT PARAMETERS TO NONDIMENSIONAL FORM AND THEN COMPUTE
    THE ZERO ORDER MOMENT AND INTERPOLATE FROM TABULATED VALUES OF
    THE HIGHER ORDER MOMENTS

INPUT
    VGRAV  THE GRAVITATIONAL SETTLING VELOCITIES OF THE PARTICLE
           IN METERS / SEC
    ZIN    THE HEIGHT (METERS) AT WHICH THE MOMENTS ARE DESIRED
    H      THE HEIGHT OF RELEASE OF THE PARTICLES IN METERS
    TIN    THE TIME IN SECONDS AFTER RELEASE

OUTPUT
    Q      THE VERTICAL CONCENTRATION OF PARTICLES AT HEIGHT Z
    XBAR   THE DISPLACEMENT (METERS) IN THE X (IE WIND) DIRECTION
           OF THE CENTER OF MASS OF PARTICLES AT HEIGHT Z
    SIGW2  THE SQUARE OF THE STANDARD DEVIATION OF THE WINDWARD
           DISPLACEMENT OF THE PARTICLES AT HEIGHT Z IN METERS**2
    SIGP2  THE SQUARE OF THE STANDARD DEVIATION OF THE CROSS-WIND
           DISPLACEMENT OF THE PARTICLES AT HEIGHT Z IN METERS**2

SUBROUTINES CALLED
    DTERPS PUTS THE NEEDED VALUES OF THE TABULATED MOMENTS
           INTO A ONE-DIMENSIONAL ARRAY
    DTERPI A FUNCTION WHICH RETURNS THE INTERPOLATED VALUE
           FOR GIVEN ARGUMENTS AND ARRAYS
    GREEN  CALCULATES THE GREENS FUNCTION WHICH IS THE
           0-ORDER MOMENT

CALLED BY CWIN
*****
IF(.NOT.FIRST)GO TO 5
READ IN THE TABLE OF MOMENTS ON THE FIRST CALL OF MOMENT
Z      LOG OF NON-DIMENSIONAL HEIGHTS AT WHICH MOMENTS ARE TABULATED
T      LOG OF NON-DIMENSIONAL TIMES AT WHICH MOMENTS ARE TABULATED
AL     NON-DIMENSIONAL SETTLING VELOCITIES AT WHICH MOMENTS ARE
      TABULATED
NM     DIFFUSIVITY POWER LAW EXPONENTS AT WHICH MOMENTS ARE
      TABULATED
XB     TABULATED VALUES OF LOGS OF FIRST ORDER MOMENTS (RELATED
      TO MEAN HORIZONTAL DISPLACEMENT)
SW     TABULATED VALUES OF LOGS OF WIND SHEAR COMPONENT OF SECOND
      ORDER MOMENT (CONTRIBUTES TO VARIANCE IN WIND DIRECTION)
SP     TABULATED VALUES OF LOGS OF SECOND ORDER MOMENT COMMON TO
      WIND AND CROSS-WIND VARIANCES

READ(NDIRTU,1) NZ,NT,NA,NH
1 FORMAT(4I3)

```

	NTC(1)=NZ-1	MOM00710
	NTC(2)=NT-1	MOM00720
	NTC(3)=NA-1	MOM00730
	NTC(4)=NN-1	MOM00740
	READ(NDIRTU,2) (Z(I),I=1,NZ)	MOM00750
2	FORMAT(6E13.5)	MOM00760
	READ(NDIRTU,2) (T(I),I=1,NT)	MOM00770
	READ(NDIRTU,2) (AL(I),I=1,NA)	MOM00780
	READ(NDIRTU,2) (NM(I),I=1,NN)	MOM00790
	NZT=NZ*NT	MOM00800
	DO 3 L=1,NN	MOM00810
	READ(NDIRTU,2) ((XB(IJ,K,L),IJ=1,NZT),K=1,NA)	MOM00820
	READ(NDIRTU,2) ((SW(IJ,K,L),IJ=1,NZT),K=1,NA)	MOM00830
	READ(NDIRTU,2) ((SP(IJ,K,L),IJ=1,NZT),K=1,NA)	MOM00840
3	CONTINUE	MOM00850
	FIRST=.FALSE.	MOM00860
	REWIND NDIRTU	MOM00870
5	CONTINUE	MOM00880
	CONVERT INPUT PARAMETERS TO NONDIMENSIONAL FORM	MOM00890
	SCLU=DZ0*H**((N-1.))	MOM00900
	XI(1)=ZIN/H	MOM00910
	XI(2)=SCLU*TIN/H	MOM00920
	XI(3)=VGRAY/SCLU	MOM00930
	XI(4)=N	MOM00940
	CALL GREEN(XI(1),HREF,XI(2),XI(3),Q,IER)	MOM00950
	Q=Q/H	MOM00960
	IF(Q .LE. 1.E-10) GO TO 999	MOM00970
	TAKE LOGS FOR LOGARITHMIC INTERPOLATION	MOM00980
	XI(1)=ALOG(XI(1))	MOM00990
	XI(2)=ALOG(XI(2))	MOM01000
	DETERMINE INDICES OF LOWEST CORNER POINT OF THE CUBE TO	MOM01010
	BE USED IN INTERPOLATION MAKING SURE THAT ENOUGH CORNER POINTS	MOM01020
	OF THE CUBE HAVE TABULATED VALUES	MOM01030
	DO 100 I=1,4	MOM01040
	II(I)=IB(I)	MOM01050
100	CONTINUE	MOM01060
	DO 101 III=1,4	MOM01070
	I=5-III	MOM01080
6	IA=II(I)	MOM01090
	IF(XI(I) .GE. X(IA,I) .AND. XI(I) .LE. X(IA+1,I)) GO TO 101	MOM01100
	IF(XI(I) .LT. X(IA,I) .AND. IA .EQ. 1) GO TO 101	MOM01110
	IF(XI(I) .GT. X(IA,I) .AND. IA .EQ. NTC(I)) GO TO 101	MOM01120
	ISAV=II(I)	MOM01130
	II(I)=IA + IFIX(SIGN(1.,XI(I)-X(IA,I)))	MOM01140
	IT=0	MOM01150
	DO 102 JI=1,2	MOM01160
	JIX=JI + II(1) - 1	MOM01170
	DO 102 IJ=1,2	MOM01180
	IJX=JIX + (IJ + II(2) - 2)*NZ	MOM01190
	DO 102 K=1,2	MOM01200
	KX=K-1 + II(3)	MOM01210
	DO 102 L=1,2	MOM01220
	LX=L-1 + II(4)	MOM01230
	IF(XB(IJX,KX,LX) .GT. -100.) IT=IT+1	MOM01240
102	CONTINUE	MOM01250
	IF(IT .GT. ITC) GO TO 6	MOM01260
	II(I)=ISAV	MOM01270
101	CONTINUE	MOM01280
	PERFORM THE INTERPOLATION WITH DETERMINED CUBE OF POINTS	MOM01290
	DO 103 I=1,4	MOM01300
	I2=I*2	MOM01310
	I1=I2-1	MOM01320
		MOM01330
		MOM01340
		MOM01350
		MOM01360
		MOM01370
		MOM01380
		MOM01390
		MOM01400

	IA=II(I)	MOM01410
	XVAL(I1)=X(IA,I)	MOM01420
	XVAL(I2)=X(IA+1,I)	MOM01430
103	CONTINUE	MOM01440
	CALL DTERPS(II,XB,VAL,NZ)	MOM01450
	XBAR=DTERRPI(4,XI,XVAL,VAL,-100.,W)	MOM01460
	CALL DTERPS(II,SW,VAL,NZ)	MOM01470
	SIGW2=DTERRPI(-4,XI,XVAL,VAL,-100.,W)	MOM01480
	CALL DTERPS(II,SP,VAL,NZ)	MOM01490
	SIGP2=DTERRPI(-4,XI,XVAL,VAL,-100.,W)	MOM01500
C		MOM01510
C	CONVERT THE LOG OF THE NONDIMENSIONAL VALUES INTERPOLATED	MOM01520
C	TO THE USUAL DIMENSIONAL FORM	MOM01530
	SCL=U0*H**((M+1.)/SCLU)	MOM01540
	XBAR=SCL*EXP(XBAR)	MOM01550
	SIGW2=SCL*SCL*EXP(SIGW2)	MOM01560
	SIGP2=2.*DX20*H*H*EXP(SIGP2)	MOM01570
	SIGW2=SIGW2+SIGP2	MOM01580
	SIGP2=DYX0*SIGP2	MOM01590
999	RETURN	MOM01600
	END	MOM01610
		MOM01620

SUBROUTINE PATH(T,U,XCEN,YCEN,RAD,PLEN)

THIS SUBROUTINE COMPUTES THE PATH LENGTH THROUGH THE SPHERE OR
WAKE FOR A HORIZONTAL PATH.

INPUTS

T - TRANSMITTER COORDINATE IN THE LOCAL COORDINATE SYSTEM
U - UNIT VECTOR ALONG THE LINE CONNECTING THE TRANSMITTER
AND RECEIVER
XCEN - X COORDINATE OF THE CENTER OF THE CIRCLE
YCEN - Y COORDINATE OF THE CENTER OF THE CIRCLE
RAD - RADIUS AT THE DESIRED HEIGHT

OUTPUT

PLEN - LENGTH OF THE INTERSECTION OF THE CONE AT HEIGHT T(3)
AND THE LINE OF SIGHT

FUNCTIONS AND SUBROUTINES NEEDED

NONE

DIMENSION T(3),U(3)
A=U(1)**2+U(2)**2
PLEN=0.
X=RAD**2*A-(U(2)*(T(1)-XCEN)-U(1)*(T(2)-YCEN))**2
IF (X.GT.0.) PLEN=2.*SQRT(X)/A
RETURN
END

PATH0290
PATH0010
PATH0020
PATH0030
PATH0040
PATH0050
PATH0060
PATH0070
PATH0080
PATH0090
PATH0100
PATH0110
PATH0120
PATH0130
PATH0140
PATH0150
PATH0160
PATH0170
PATH0180
PATH0190
PATH0200
PATH0210
PATH0220
PATH0230
PATH0240
PATH0250
PATH0260
PATH0270
PATH0280
PATH0300
PATH0310

PATH0400


```
      SUBROUTINE PERP(A,B)
      DIMENSION A(2),B(2)
C *** B IS ROTATED 90 DEGREES COUNTERCLOCKWISE FROM A
      B(1)=-A(2)
      B(2)=A(1)
      RETURN
      END
```

```
PER00010
PER00020
PER00030
PER00040
PER00050
PER00060
PER00070
```

```

SUBROUTINE PRECL(NATMOS,ZTEMP,TMPMES,ZWND,WNDMES,THWND,PHI,DHDT,
1 CHWT,NCHRG,DETDEP,NSOIL,DSOD,SILT)
ROUTINE TO PRECOMPUTE EXPLOSION PRODUCED DUST CLOUD AND STORE ON
EXTERNAL FILE UNIT IFILE
OUTPUTS
  XC1 COEFFICIENTS OF QUADRATIC FIT TO SLOPE OF LINE DESCRIBING
    X DISPLACEMENT OF CONE (ALONGWIND)
  XC0 COEFFICIENTS OF QUADRATIC FIT TO CONSTANT TERM OF LINE
    DESCRIBING X DISPLACEMENT OF CONE (ALONGWIND)
  Z2 COEFFICIENTS OF QUADRATIC FIT TO HEIGHT OF TOP OF CLOUD
  RT COEFFICIENTS OF QUADRATIC FIT TO THE RADIUS OF THE TOP OF THE
    CONE
  RB COEFFICIENTS OF THE QUADRATIC FIT TO THE RADIUS OF THE CONE
    AT A HEIGHT OF THE AVERAGE OF THE DISC SOURCES

THE ABOVE OUTPUT ARE THE COEFFICIENTS OF QUADRATIC FITS THROUGH
THREE CONSECUTIVE POINT IN TIME. THE QUADRATIC FITS ARE STORED IN
COMMON/ MOS / WITH THE ARRAY TIMES CONTAINING THE LAST TIME
OF EACH QUADRATIC PIECE WITH THE FIRST STARTING AT 0.0
THE FITS ARE WRITTEN ONTO A FILE INDICATED BY IFILE USING
A BINARY WRITE
THE FITS ARE STORED SUCH THAT
  F(TIME)=VAR(1,J)*TIME**2 + VAR(2,J)*TIME + VAR(3,J)
  AND TIMES(J-1) < TIME < TIMES(J)
*****
LOGICAL SWITCH,CHANGE,DHDT
REAL KZ,KX
DIMENSION T(3),FRB(3),FRT(3),FXC1(3),FXC0(3),XB(3),OWF(5,2)
DIMENSION FZ2(3),ZTEMP(2),TMPMES(2),ZWND(2),WDMES(2),OWFC(5)
COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLINT,KSTOR,NPLOTU
COMMON/OPTION/IOPT,IFILE
COMMON/DISCS/NDSCS,TDSC(20),XDSC(20),ZDSC(20),R2DSC(20),QDSC(20,3)
COMMON/BUQYCL/RSPH,DELT,VZ,XCM,YCM,ZCM,XTOP,YTOP,SPHNS(3),TIM
COMMON/MOS/DMMY(604),DMM(600),
+ ICOUNT,TIMES(25),XC0(3,25),XC1(3,25),RT(3,25),
+ RB(3,25),Z2(3,25)
COMMON/VL/VLOAD
COMMON/SIG/SIG02,SIGC
COMMON/CLOCK/FTIME,TWIND
COMMON/TRAN/VTR,KZ,KX,TTR,XTR,ZTR,QPUFF(3),SWITCH,CHANGE
COMMON/WNDPRM/DX20,DYX0,DZ0,U0,UM,DN,ZINV
COMMON/GEOM/COSTH2,SINTH,SINTH2,VISEXT,RTPI,SCRN(2)
COMMON/CARB/RCARB1,RCARB2
COMMON /CONST/P1,P12,PIRAD,TWOPI,TORRMB,CDEGK
DATA OWF/1.,.93,.52,.44,2.E-03,1.,1.,1.,1.,.4.E-03/
DATA OWFC/1.,.95,.5,.2,1.E-03/
TINC=1.0
TPRES=0.0
ICOUNT=0
DEL=.001
T(3)=0.0
SUM2=0.0

FIND AVERAGE OF THE DISC RELEASE HEIGHTS AND THE AVERAGE OF THE
INITIAL SPREADS OF THE DISCS
  DO 5 J=1,NDSCS
  SUM2=SUM2+R2DSC(J)
5 CONTINUE

```

```

PRE00350
PRE00360
PRE00010
PRE00020
PRE00030
PRE00040
PRE00050
PRE00060
PRE00070
PRE00080
PRE00090
PRE00100
PRE00110
PRE00120
PRE00130
PRE00140
PRE00150
PRE00160
PRE00170
PRE00180
PRE00190
PRE00200
PRE00210
PRE00220
PRE00230
PRE00240
PRE00250
PRE00260
PRE00270
PRE00280
PRE00290
PRE00300
PRE00310
PRE00320
PRE00330
PRE00340
PRE00370
PRE00380
PRE00390
PRE00400
PRE00410
PRE00420
PRE00430
PRE00440
PRE00450
PRE00460
PRE00470
PRE00480
PRE00490
PRE00500
PRE00510
PRE00520
PRE00530
PRE00540
PRE00550
PRE00560
PRE00570
PRE00580
PRE00590
PRE00600
PRE00610
PRE00620
PRE00630
PRE00640
PRE00650
PRE00660
PRE00670
PRE00680
PRE00690
PRE00700

```

	ZREF=2.0	PRE 00710
	ROH2=SUM2/NDSCS	PRE 00720
10	ICOUNT=ICOUNT+1	PRE 00730
	DO 50 I=1,3	PRE 00740
	T(I)=T(3)+FLOAT(I-1)*TINC	PRE 00750
C		PRE 00760
C	FIND THE AVERAGE OF THE MOMENTS AT HEIGHT ZREF	PRE 00770
	CALL AVRG(ZREF,T(I),QTOT,XBAVRG,SIG2X,SIG2Y)	PRE 00780
	IF(QTOT.LT.1.E-10)GO TO 15	PRE 00790
	SIGX=SQRT(SIG2X+ROH2/2.)	PRE 00800
	SIGY=SQRT(SIG2Y+ROH2/2.)	PRE 00810
	FRB(I)=SQRT(SIGX*SIGY)*1.5	PRE 00820
	XB(I)=XBAVRG	PRE 00830
	GO TO 20	PRE 00840
15	CALL WIN(ZREF,UX,V)	PRE 00850
	XB(I)=UX*T(I)	PRE 00860
	FRB(I)=0.0	PRE 00870
20	IF(T(I).GT.TWIND)GO TO 30	PRE 00880
	CALL RISE(TPRES,T(I),DEL)	PRE 00890
	FZ2(I)=ZCM+(2./3.)*RSPH	PRE 00900
	IF(FZ2(I).GT.ZINV)FZ2(I)=ZINV	PRE 00910
	FRT(I)=RSPH	PRE 00920
	XB(2)=XCM	PRE 00930
	GO TO 40	PRE 00940
30	XB(2)=XTR+VTR*(T(I)-TTR)	PRE 00950
	SIGX2=SIG02+2.*KX*(T(I)-TTR)	PRE 00960
	SIGZ2=SIG02+2.*KZ*(T(I)-TTR)	PRE 00970
	SIGX=SQRT(SIGX2)	PRE 00980
	SIGZ=SQRT(SIGZ2)	PRE 00990
	SIG=SQRT(SIGX*SIGZ)	PRE 01000
	FZ2(I)=ZTR+SIG	PRE 01010
	FRT(I)=1.5*SIG	PRE 01020
40	FXC(I)=(XB(2)-XB(1))/(FZ2(I)-ZREF)	PRE 01030
	FXC0(I)=XB(1)-FXC(I)*ZREF	PRE 01040
50	CONTINUE	PRE 01050
C		PRE 01060
C	COMPUT AND STORE QUADRATIC FITS	PRE 01070
	TIMES(ICOUNT)=T(3)	PRE 01080
C		PRE 01090
C	FIT AND STORE RADIUS AT TOP	PRE 01100
	CALL FIT(T,FRT,A,B,C)	PRE 01110
	RT(1,ICOUNT)=A	PRE 01120
	RT(2,ICOUNT)=B	PRE 01130
	RT(3,ICOUNT)=C	PRE 01140
C		PRE 01150
C	FIT AND STORE RADIUS AT BOTTOM	PRE 01160
	CALL FIT(T,FRB,A,B,C)	PRE 01170
	RB(1,ICOUNT)=A	PRE 01180
	RB(2,ICOUNT)=B	PRE 01190
	RB(3,ICOUNT)=C	PRE 01200
C		PRE 01210
C	FIT AND STORE HEIGHT OF CLOUD	PRE 01220
	CALL FIT(T,FZ2,A,B,C)	PRE 01230
	Z2(1,ICOUNT)=A	PRE 01240
	Z2(2,ICOUNT)=B	PRE 01250
	Z2(3,ICOUNT)=C	PRE 01260
C		PRE 01270
C	FIT AND STORE XC1	PRE 01280
	CALL FIT(T,FXC1,A,B,C)	PRE 01290
	XC1(1,ICOUNT)=A	PRE 01300
	XC1(2,ICOUNT)=B	PRE 01310
	XC1(3,ICOUNT)=C	PRE 01320
C		PRE 01330
C	FIT AND STORE XC0	PRE 01340
		PRE 01350
		PRE 01360
		PRE 01370
		PRE 01380
		PRE 01390
		PRE 01400

C	CALL FIT(T,FXC0,A,B,C)	PRE 01410
	XC0(1,ICOUNT)=A	PRE 01420
	XC0(2,ICOUNT)=B	PRE 01430
	XC0(3,ICOUNT)=C	PRE 01440
C	CHECK TO SEE IF PRECOMPUTE CAN BE STOPPED	PRE 01450
C	IF(FRT(3).GE.FRB(3))PLEN=2.*FRT(3)	PRE 01460
C	IF(FRB(3).GE.FRT(3))PLEN=2.*FRB(3)	PRE 01470
	VOL=(PI/3.)*(FZ2(3)-ZREF)*(FRT(3)**2+FRT(3)*FRB(3)+FRB(3)**2)	PRE 01480
	NWL=1	PRE 01490
	ACL=(RCARB1*OWF(NWL,NSOIL)+RCARB2*OWF(NWL))*VLOAD*PLEN/VOL	PRE 01500
	TINC=1.2*TINC	PRE 01510
	IF(ICOUNT.LT.25.AND.ACL.GT..001)GO TO 10	PRE 01520
C	WRITE PRECOMPUTED CLOUD INFORMATION ONTO FORTRAN UNIT IFILE WITH	PRE 01530
C	A BINARY WRITE	PRE 01540
	WRITE(IFILE)NATMOS,ZTEMP(1),TMPMES(1),ZWND(1),WNDMES(1)	PRE 01550
	WRITE(IFILE)DHDT,PHI,CHWT,NCHRG,DETDEP,NSOIL,DSOD,SILT,ZINV	PRE 01560
	WRITE(IFILE)VLOAD,RCARB1,RCARB2	PRE 01570
	WRITE(IFILE)ICOUNT	PRE 01580
	DO 1000 J=1,ICOUNT	PRE 01590
	WRITE(IFILE) TIMES(J),(RT(1,J),RB(1,J),Z2(1,J),XC0(1,J),	PRE 01600
	XC1(1,J),I=1,3)	PRE 01610
1000	CONTINUE	PRE 01620
	REWIND IFILE	PRE 01630
999	RETURN	PRE 01640
	END	PRE 01650
		PRE 01660
		PRE 01670
		PRE 01680
		PRE 01690
		PRE 01700

```

C      SUBROUTINE PRETRN(TRN,REC,TIME,TRNLOS)
C      COMPUTE THE TRANSMITTANCE FOR THE RANDOM IN SPACE AND TIME
C      DISTRIBUTION OF CHARGES
C      INPUTS
C      TRN  -TRANSMITTER COORDINATES IN LOCAL COORDINATE SYSTEM
C      REC  -RECEIVER COORDINATES IN LOCAL COORDINATE SYSTEM
C      TIME -TIME AT WHICH TRANSMITTANCE IS DESIRED
C      NWL  -INTEGER INDEX FOR WAVELENGTH
C      NSOIL -SOIL TYPE
C      OUTPUTS
C      TRNLOS -TRANSMITTANCE ALONG THE SPECIFIED LINE OF SIGHT
C      *****
C      LOGICAL TEST
C      DIMENSION TRN(3),REC(3),TR(3),RE(3),OWF(5,2),U(3),OWFC(5)
C      COMMON/MO5/DIFF(2,200),NCHTOT,PRSEP(200),WTOT,NARY,ITOT,
C      +      COOR(2,200),TSTAG(200),
C      +      ICOUNT,TIMES(25),XC0(3,25),XC1(3,25),RT(3,25),
C      +      RB(3,25),Z2(3,25)
C      COMMON/VL/VLOAD
C      COMMON/TRANNY/THRESH,TEST,NWL,NSOIL
C      COMMON/CARB/RCARB1,RCARB2
C      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK
C      DATA OWF/1.,.93,.52,.44,2.E-03,1.,1.,1.,1.,4.E-03/
C      DATA OWFC/1.,.95,.5,.2,1.E-03/
C      PARAMETERIZE THE LINE CONNECTING THE TRANSMITTER AND RECEIVER
C      TEST=.FALSE.
C      XNORM=0.0
C      DO 10 I=1,3
C      RE(I)=REC(I)
C      TR(I)=TRN(I)
C      U(I)=RE(I)-TR(I)
C      XNORM=XNORM+U(I)**2
C 10 CONTINUE
C      XNORM=SQRT(XNORM)
C      U(1)=U(1)/XNORM
C      U(2)=U(2)/XNORM
C      U(3)=U(3)/XNORM
C      COMPUTE THE CONTRIBUTION FROM EACH CHARGE TO THE OPTICALLY
C      WEIGHTED CONCENTRATION ALONG THE LINE OF SIGHT
C      SUM=0.0
C      DO 100 I=1,ITOT
C      IF(TIME.LT.TSTAG(I))GO TO 100
C      TOF=TIME-TSTAG(I)
C      IF(TOF.GT.TIMES(ICOUNT))GO TO 100
C      DO 20 J=1,ICOUNT
C      IND=J
C      IF(TOF.LE.TIMES(J))GO TO 30
C 20 CONTINUE
C      DETERMINE NECESSARY PARAMETERS DESCRIBING THE CONICAL SHAPE SO THAT
C      THE LENGTH OF INTERSECTION OF THE LINE OF SIGHT AND CONE CAN BE
C      DETERMINED
C 30 X0=TOF*(XC0(1,IND)*TOF+XC0(2,IND))+XC0(3,IND)
C      X1=TOF*(XC1(1,IND)*TOF+XC1(2,IND))+XC1(3,IND)
C      HTTOP=TOF*(Z2(1,IND)*TOF+Z2(2,IND))+Z2(3,IND)

```

```

PRT00220
PRT00010
PRT00020
PRT00030
PRT00040
PRT00050
PRT00060
PRT00070
PRT00080
PRT00090
PRT00100
PRT00110
PRT00120
PRT00130
PRT00140
PRT00150
PRT00160
PRT00170
PRT00180
PRT00190
PRT00200
PRT00210
PRT00230
PRT00240
PRT00250
PRT00260
PRT00270
PRT00280
PRT00290
PRT00300
PRT00310
PRT00320
PRT00330
PRT00340
PRT00350
PRT00360
PRT00370
PRT00380
PRT00390
PRT00400
PRT00410
PRT00420
PRT00430
PRT00440
PRT00450
PRT00460
PRT00470
PRT00480
PRT00490
PRT00500
PRT00510
PRT00520
PRT00530
PRT00540
PRT00550
PRT00560
PRT00570
PRT00580
PRT00590
PRT00600
PRT00610
PRT00620
PRT00630
PRT00640
PRT00650
PRT00660
PRT00670
PRT00680
PRT00690
PRT00700

```

	RTOP=TOF*(RT(1,IND)*TOF+RT(2,IND))+RT(3,IND)	PRT00710
	HTBOT=0.0	PRT00720
	RBOT=TOF*(RB(1,IND)*TOF+RB(2,IND))+RB(3,IND)	PRT00730
	XCEN=DIFF(1,1)+(X1*HTTOP+X0)	PRT00740
	YCEN=DIFF(2,1)	PRT00750
	XB=DIFF(1,1)+(X1*HTBOT+X0)	PRT00760
	YB=DIFF(2,1)	PRT00770
	IF(ABS(U(3)),LT.1.E-06)GO TO 40	PRT00780
C	COMPUTE THE INTERSECTION LENGTH FOR A NON-HORIZONTAL LINE OF SIGHT	PRT00790
C	CALL CONLEN(U,TR,HTTOP,HTBOT,XCEN,YCEN,RTOP,RBOT,XB,YB,	PRT00800
C	1 GO TO 50 XNORM,PLEN)	PRT00810
C	COMPUTE THE INTERSECTION LENGTH FOR A HORIZONTAL LINE OF SIGHT	PRT00820
C	40 IF(HTTOP,LT,TR(3))GO TO 45	PRT00830
C	A=U(1)**2+U(2)**2	PRT00840
C	B=U(1)*(TR(1)-XCEN)+U(2)*(TR(2)-YCEN)	PRT00850
C	DETERMINE THE RADIUS,X,AND Y POSITIONS OF THE CONE AT THE	PRT00860
C	TRANSMITTER HEIGHT	PRT00870
C	ZETA=TR(3)/HTTOP	PRT00880
	RAD=ZETA*RTOP+(1.-ZETA)*RBOT	PRT00890
	XCEN=ZETA*XCEN+(1.-ZETA)*XB	PRT00900
	YCEN=ZETA*YCEN+(1.-ZETA)*YB	PRT00910
	A=U(1)**2+U(2)**2	PRT00920
	B=U(1)*(TR(1)-XCEN)+U(2)*(TR(2)-YCEN)	PRT00930
	C=(TR(1)-XCEN)**2+(TR(2)-YCEN)**2-RAD**2	PRT00940
	X=B**2-A*C	PRT00950
	IF(X,LT,0.0)GO TO 45	PRT00960
	P1=(-B+SQRT(X))/A	PRT00970
	P2=(-B-SQRT(X))/A	PRT00980
	IF(P1.GT.XNORM.AND,P2.GT.XNORM)GO TO 45	PRT00990
	IF(P2,LT,0.0.AND,P1,LT,0.0)GO TO 45	PRT01000
	PLEN=AMIN1(P1,XNORM)-AMAX1(P2,0.0)	PRT01010
	GO TO 50	PRT01020
45	PLEN=0.0	PRT01030
50	VOL=(PI/3.)*(HTTOP-HTBOT)*(RTOP**2+RTOP*RBOT+RBOT**2)	PRT01040
	CONT=VLOAD*PLEN/VOL	PRT01050
	SUM=SUM+CONT	PRT01060
	ACLSKT=0.0	PRT01070
	ACLSPH=0.0	PRT01080
	CALL TRNCHK(ACLSKT,SUM,ACLSPH)	PRT01090
	IF(TEST)GO TO 998	PRT01100
100	CONTINUE	PRT01110
	TRNLOS=EXP(-SUM*(RCARB1*OWF(NWL,NSOIL)+RCARB2*OWF(NWL)))	PRT01120
	GO TO 999	PRT01130
998	TRNLOS=0.0	PRT01140
999	RETURN	PRT01150
	END	PRT01160
		PRT01170
		PRT01180
		PRT01190
		PRT01200
		PRT01210
		PRT01220
		PRT01230

```

SUBROUTINE PREVEH(NSOIL,NWL)
ROUTINE FOR PRECOMPUTING VEHICLE GENERATED DUST CLOUD

INPUTS
  NSOIL - SOIL TYPE (SEE DRTRAN FOR DETAILS)
  NWL   - WAVELENGTH INDEX (SEE DRTRAN)

OUTPUTS
  RT - COEFFICIENTS OF QUADRATIC FIT TO RADIUS OF THE CLOUD
  Z2 - COEFFICIENTS OF QUADRATIC FIT TO HEIGHT OF THE CLOUD
  XC1 - COEFFICIENTS OF QUADRATIC FIT TO SLOPE OF LINE DESCRIBING
        X DISPLACEMENT OF CONE (ALONGWIND)
  XC0 - COEFFICIENTS OF QUADRATIC FIT TO CONSTANT TERM OF LINE
        DESCRIBING X DISPLACEMENT OF THE CONE (ALONGWIND)

THE ABOVE OUTPUT ARE THE COEFFICIENTS OF QUADRATIC FITS THROUGH
THREE CONSECUTIVE POINTS IN TIME. THE QUADRATIC FITS ARE STORED IN
COMMON/ M05 / WITH THE ARRAY TIMES CONTAINING THE LAST TIME OF THAT
INTERVAL. THE FITS ARE WRITTEN ONTO A FILE INDICATED BY IFILE USING
A BINARY WRITE.

THE FITS ARE STORED SUCH THAT
  F(TIME)=VAR(1,J)*TIME**2 + VAR(2,J)*TIME + VAR(3,J)
AND
  TIMES(J-1) < TIME < TIMES(J)
C*****
LOGICAL HORIZ
DIMENSION OWF(5,2),T(3),FR(3),FZ2(3)
DIMENSION FXC(3),FXC0(3),XB(2)
COMMON/DISCS/NDSCS,TDSC(20),XDSC(20),ZDSC(20),R2DSC(20),QDSC(20,3)
COMMON/PRE/ZT0,RT2DZ
COMMON/MODE/HORIZ
COMMON/M05/DIFF(2,200),NCHTOT,PRSEP(200),NTOT,NARY,ITOT,DMM(600),
+ ICOUNT,TIMES(25),XC0(3,25),XC1(3,25),RT(3,25),
+ RB(3,25),Z2(3,25)
COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NBIRTU,NCLIMT,KSTOR,NPLOTUP
COMMON/PRTINF/R0,VGRAY(3),NPRTS
COMMON/GEOM/COSTH2,SINTH,SINTH2,VISEXT,RTPI,SCRN(2)
COMMON /CONST/PI,P12,PIRAD,TWOPI,TORRMB,CDEGK
C
DATA OWF/1.,.93,.52,.44,2.E-03,1.,1.,1.,1.,4.E-03/
C
SET UP CORRECT PARAMETERS SO THAT CWINDD CAN BE USED
C
NARY=1
ITOT=1
NTOT=1
NCHTOT=1
HORIZ=.TRUE.
ONEM=-1.
COSTH=1.0
SINTH=0.0
COSTH2=COSTH**2
SINTH2=SINTH**2
SCRN(1)=SINTH
SCRN(2)=-COSTH
X=0.0
Y=0.0
TMAX=1000.

```

```

PRV00360
PRV00010
PRV00020
PRV00030
PRV00040
PRV00050
PRV00060
PRV00070
PRV00080
PRV00090
PRV00100
PRV00110
PRV00120
PRV00130
PRV00140
PRV00150
PRV00160
PRV00170
PRV00180
PRV00190
PRV00200
PRV00210
PRV00220
PRV00230
PRV00240
PRV00250
PRV00260
PRV00270
PRV00280
PRV00290
PRV00300
PRV00310
PRV00320
PRV00330
PRV00340
PRV00350
PRV00370
PRV00380
PRV00390
PRV00400
PRV00410
PRV00420
PRV00430
PRV00440
PRV00450
PRV00460
PRV00470
PRV00480
PRV00490
PRV00500
PRV00510
PRV00520
PRV00530
PRV00540
PRV00550
PRV00560
PRV00570
PRV00580
PRV00590
PRV00600
PRV00610
PRV00620
PRV00630
PRV00640
PRV00650
PRV00660
PRV00670
PRV00680
PRV00690
PRV00700

```

C	COMPUTE R, Z2, XC1, XC0 AT THREE CONSECUTIVE TIMES WITH SPACING	PRV00710
CC	TINC AND THEN CALL FIT, WHICH CALCULATES A QUADRATIC FIT TO THESE	PRV00720
CC	POINTS AND STORE THEM IN COMMON /QUADFT/	PRV00730
C		PRV00740
	TINC=1.0	PRV00750
	ICOUNT=0	PRV00760
	T(3)=0.0	PRV00770
10	ICOUNT=ICOUNT+1	PRV00780
	DO 20 I=1,3	PRV00790
	T(I)=T(3)+FLOAT(I-1)*TINC	PRV00800
	ZREF=ZT0+SQRT(T(I))*RT2DZ	PRV00810
	TOF=T(I)-TDSC(NDSCS)	PRV00820
	CALL MOMENT(VGRAV,ZREF,ZDSC(I),TOF,Q,XBAR,SIGW2,SIGP2)	PRV00830
C	COMPUTE R THE RADIUS OF THE CLOUD	PRV00840
CC		PRV00850
	ROH2=R2DSC(I)	PRV00860
	SIGW=SQRT(SIGW2+ROH2/2.)	PRV00870
	SIGP=SQRT(SIGP2+ROH2/2.)	PRV00880
	FR(I)=1.5*SQRT(SIGW*SIGP)	PRV00890
	ACL=CWIND(X,Y,ZREF,T(I))*OWF(NWL,NSOIL)	PRV00900
C	COMPUTE Z2 APPROXIMATE HEIGHT OF THE CLOUD	PRV00910
CC		PRV00920
	FZ2(I)=(2.*QDSC(1,1))/PI/FR(I)/ACL	PRV00930
	Z22=FZ2(I)	PRV00940
C	COMPUTE THE X POSITION OF THE CLOUD AT A HEIGHT OF Z2 AND A HEIGHT	PRV00950
CC	OF 1 METER.	PRV00960
	CALL MOMENT(VGRAV,Z22,ZDSC(I),TOF,Q,XBAR,SIGW2,SIGP2)	PRV00970
	XB(2)=XBAR+XDSC(1)	PRV00980
	Z1=1.0	PRV00990
	CALL MOMENT(VGRAV,Z1,ZDSC(1),TOF,Q,XBAR,SIGW2,SIGP2)	PRV01000
	XB(1)=XBAR+XDSC(1)	PRV01010
	FXC1(I)=(XB(2)-XB(1))/(Z22-Z1)	PRV01020
	FXC0(I)=XB(1)-FXC1(I)	PRV01030
20	CONTINUE	PRV01040
C	COMPUTE AND STORE THE QUADRATIC FITS	PRV01050
CC		PRV01060
	TIMES(ICOUNT)=T(3)	PRV01070
C	FIT AND STORE THE CLOUD RADIUS	PRV01080
CC		PRV01090
	CALL FIT(T,FR,A,B,C)	PRV01100
	RT(1,ICOUNT)=A	PRV01110
	RT(2,ICOUNT)=B	PRV01120
	RT(3,ICOUNT)=C	PRV01130
C	FIT AND STORE Z2, APPROXIMATE CLOUD HEIGHT	PRV01140
CC		PRV01150
	CALL FIT(T,FZ2,A,B,C)	PRV01160
	Z2(1,ICOUNT)=A	PRV01170
	Z2(2,ICOUNT)=B	PRV01180
	Z2(3,ICOUNT)=C	PRV01190
C	FIT AND STORE XC1	PRV01200
CC		PRV01210
	CALL FIT(T,FXC1,A,B,C)	PRV01220
	XC1(1,ICOUNT)=A	PRV01230
	XC1(2,ICOUNT)=B	PRV01240
	XC1(3,ICOUNT)=C	PRV01250
C	FIT AND STORE XC0	PRV01260
CC		PRV01270
	CALL FIT(T,FXC0,A,B,C)	PRV01280
	XC0(1,ICOUNT)=A	PRV01290
	XC0(2,ICOUNT)=B	PRV01300
	XC0(3,ICOUNT)=C	PRV01310
C		PRV01320
CC		PRV01330
		PRV01340
		PRV01350
		PRV01360
		PRV01370
		PRV01380
		PRV01390
		PRV01400

TINC=1.2*TINC
IF<ICOUNT.LT.20.AND.T(3).LT.TMAX> GO TO 10
RETURN
END

PRV01410
PRV01420
PRV01430
PRV01440

```

SUBROUTINE RISE(TPRES,TNEXT,DEL)
REAL M,NDIF,KZ,KX
LOGICAL SWITCH,CHANGE
DIMENSION WK(12,6)
COMMON/BUOYCL/ Y(8),SPHNS(3),RISTIM
COMMON/ WNDPRM/ DXZ0,DYX0,DZ0,U0,M,NDIF,ZINV
COMMON/ CLOCK/ TIME,TWIND
COMMON/ STARS/ USTAR,TSTAR,ZSTAR
COMMON/ EKTEMP/ Z0,ZL,T0,TC1,TC2,TC3
COMMON/ TRAN/ VTR,KZ,KX,TTR,XTR,ZTR,QPUFF(3),SWITCH,CHANGE
COMMON/ SIG/ SIG02,SIGC
COMMON/ IOUNIT/ IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUR
COMMON/ DISCS/ NDSCS,TDSC(20),XDSC(20),ZDSC(20),R2DSC(20),
1QDSC(20,3)
COMMON/ BURST/ ACCEL,TBURST
DATA HMIN,ACCURC,WK,N,ND/.001,.001,72*0.,8,12/

```

PURPOSE

THIS ROUTINE CALLS A RUNGA KUTTA ROUTINE TO INTEGRATE IN TIME THE EQUATIONS FOR THE RISE OF A BUOYANT CLOUD BEGINNING AT TPRES ENDING AT TNEXT UNLESS THE CONDITION FOR SWITCHING TO THE WIND DISPERSION MODEL IS ENCOUNTERED IN WHICH CONVRT IS CALLED. SEE SUBROUTINE DIFEQ FOR THE DEFINITIONS OF Y(I).

ARGUMENTS

TPRES AS INPUT TPRES IS THE INITIAL TIME OF THIS SEGMENT OF INTEGRATION AND IS RETURNED WITH THE VALUE OF THE LAST SUCCESSFUL INTEGRATION STEP.

TNEXT THE ENDPOINT OF THE TIME INTERVAL WHICH IS INPUT.

REQUIRED SUBROUTINES

RKM A RUNGA-KUTTA-MERSON INTEGRATION ROUTINE

CONVRT A SUBROUTINE WHICH CONVERTS THE CURRENT BUOYANT DUST CLOUD TO A NUMBER OF DISC SOURCES FOR THE WIND DISPERSION MODEL. A GAP TIME DURING WHICH THE BUOYANT MODEL IS CONTINUED IS COMPUTED.

WINDCAL COMPUTES SCALED WIND SPEED AT A SPECIFIED HEIGHT

DIFFUS COMPUTES DIFFUSIVITY AT A SPECIFIED HEIGHT

CALLED BY DUSTCL

```

IF(TNEXT.GT.TWIND)GO TO 999
SWITCH=.FALSE.
CHANGE=.FALSE.
T2=TPRES

```

PERFORM INTEGRATION IN SEGMENTS OF TIME

```

10 DO 20 NT=1,300
T1=T2
T2=1.2*T1
IF(T2.LE.0.)T2=.5
IF(T2.GT.TNEXT)T2=TNEXT
IF (DEL.LT.HMIN)DEL=HMIN
CALL RKM(N,T1,T2,Y,HMIN,DEL,ACCURC,WK,ND)

```

CHECK TO SEE IF CLOUD GROWTH IS DOMINATED BY WIND DIFFUSION OVER BUOYANT RISE BY COMPARING WIND DIFFUSIVITY, DIFW, TO THE EFFECTIVE BUOYANT DIFFUSIVITY, DIFB AND IF THE HEIGHT OF THE CENTER OF MASS IS LESS THAN ZSTAR SWITCH TO THE

RIS00010
RIS00020
RIS00030
RIS00040
RIS00050
RIS00060
RIS00070
RIS00080
RIS00090
RIS00100
RIS00110
RIS00120
RIS00130
RIS00140
RIS00150
RIS00160
RIS00170
RIS00180
RIS00190
RIS00200
RIS00210
RIS00220
RIS00230
RIS00240
RIS00250
RIS00260
RIS00270
RIS00280
RIS00290
RIS00300
RIS00310
RIS00320
RIS00330
RIS00340
RIS00350
RIS00360
RIS00370
RIS00380
RIS00390
RIS00400
RIS00410
RIS00420
RIS00430
RIS00440
RIS00450
RIS00460
RIS00470
RIS00480
RIS00490
RIS00500
RIS00510
RIS00520
RIS00530
RIS00540
RIS00550
RIS00560
RIS00570
RIS00580
RIS00590
RIS00600
RIS00610
RIS00620
RIS00630
RIS00640
RIS00650
RIS00660
RIS00670
RIS00680
RIS00690
RIS00700

C	WIND MODEL.	RIS00710
C		RIS00720
5	IF(T2.LT.TBURST)GO TO 15	RIS00730
	DIFB=ABS(.1111*Y(1)*Y(3))	RIS00740
	TOP=Y(6)+Y(1)	RIS00750
	DIFW=DIFFUS(Z0,ZL, TOP)	RIS00760
	IF(DIFB.GT.DIFW)GO TO 15	RIS00770
	IF(TOP.GT.ZSTAR)SWITCH=.TRUE.	RIS00780
	IF(TOP.GT.ZSTAR)GO TO 15	RIS00790
	VTR=USTAR*WINDCAL(Z0,ZL,Y(6))	RIS00800
	KZ=DIFFUS(Z0,ZL,Y(6))	RIS00810
	KX=DXZ0*KZ	RIS00820
	TTR=T2	RIS00830
	XTR=Y(4)	RIS00840
	ZTR=Y(6)	RIS00850
	CHANGE=.TRUE.	RIS00860
	CALL CONVRT(T2)	RIS00870
	GO TO 200	RIS00880
15	CONTINUE	RIS00890
	IF(T2.GE.TNEXT)GO TO 200	RIS00900
	IF(T2.GT.300.)GO TO 99	RIS00910
20	CONTINUE	RIS00920
99	WRITE(100OUT,98)	RIS00930
98	FORMAT(54H *** DIRTRAN ERROR - 5 MINUTE CUT-OFF ON BUOYANT RISE)	RIS00940
	STOP	RIS00950
200	TPRES=T2	RIS00960
	RISTIM=TPRES	RIS00970
999	RETURN	RIS00980
	END	RIS00990

```

SUBROUTINE RKM(N,XL,XU,Y,HMIN,DEL,ACCURC,WK,ND)
NUMERICAL INTEGRATION ROUTINE FOR SYSTEMS OF ODE'S
  USING THE RUNGE-KUTTA-MERSON TECHNIQUE

INPUT PARAMETERS
  N - NUMBER OF FIRST ORDER DIFFERENTIAL EQUATIONS
  XL - INITIAL ABSCISSA OF THE INTERVAL
  XU - THE FINAL ABSCISSA OF THE INTEGRATION INTERVAL
  Y - A SINGLY DIMENSIONED ARRAY OF LENGTH N. WHEN
      RKM IS CALLED IT MUST CONTAIN THE VALUES OF
      THE DEPENDENT VARIABLES AT XL. UPON RETURN
      TO THE CALLING PROGRAM Y CONTAINS THE VALUES
      OF THE DEPENDENT VARIABLES AT XU.
  HMIN - THE MINIMUM STEP SIZE THAT WILL BE USED FOR THE
      INTEGRATION
  DEL - THE INITIAL ESTIMATE OF THE STEP SIZE AND UPON
      RETURN TO THE CALLING PROGRAM DEL CONTAINS THE
      FINAL STEP SIZE USED. THIS VALUE SHOULD BE USED
      IN THE NEXT CALL TO PRODUCE AN EFFICIENT INTEGRATION.
      DEL IS RETURNED WITH THE VALUE ZERO IF IT HAS
      BEEN HALVED BELOW HMIN.
  ACCURC - PREASSIGNED ACCURACY WHICH IS ALSO USED IN ADJUSTING
      THE STEP SIZE.
  WK - AT LEAST A BLOCK OF N BY 6 FLOATING POINT LOCATIONS
      USED FOR A WORK ARRAY.
  ND - THE DIMENSION OF ARRAYS Y AND WK.

IT IS REQUIRED THAT THE USER OF RKM WRITE A SUBROUTINE
DEFINING THE DIFFERENTIAL EQUATIONS. THE SUBROUTINE
STATEMENT SHOULD LOOK LIKE - SUBROUTINE DIFEQ(N,X,Y,YP) .

WHERE
  N - THE NUMBER OF EQUATIONS
  X - THE INDEPENDENT VARIABLE
  Y - SINGLY DIMENSIONED ARRAY OF DEPENDENT VARIABLES
  YP - SINGLY DIMENSIONED ARRAY OF THE RATES OF Y AT X
      YP(I) = D Y(I)/DX

  DIMENSION Y(ND),WK(ND,6)
  LOGICAL FIRST,QUIT

SET UP NEEDED VARIABLES UPON ENTRY
  XN=XL
  H=DEL
  FIRST=.TRUE.
  QUIT=.FALSE.

CHECK IF XN IS CLOSE TO XU
20 IF(XN+H .LT. XU) GO TO 30
  DEL=H
  H=XU-XN
  QUIT=.TRUE.
  IF(FIRST) DEL=H

MAKE FIRST CALL TO DIFEQ AT THE BEGINNING OF INTERVAL
30 CALL DIFEQ(N,XN,Y,WK(1,1))

PERFORM THE RUNGE-KUTTA-MERSON ALGORITHM
40 H3=H/3.
  DO 50 I=1,N

```

```

RKM00010
RKM00020
RKM00030
RKM00040
RKM00050
RKM00060
RKM00070
RKM00080
RKM00090
RKM00100
RKM00110
RKM00120
RKM00130
RKM00140
RKM00150
RKM00160
RKM00170
RKM00180
RKM00190
RKM00200
RKM00210
RKM00220
RKM00230
RKM00240
RKM00250
RKM00260
RKM00270
RKM00280
RKM00290
RKM00300
RKM00310
RKM00320
RKM00330
RKM00340
RKM00350
RKM00360
RKM00370
RKM00380
RKM00390
RKM00400
RKM00410
RKM00420
RKM00430
RKM00440
RKM00450
RKM00460
RKM00470
RKM00480
RKM00490
RKM00500
RKM00510
RKM00520
RKM00530
RKM00540
RKM00550
RKM00560
RKM00570
RKM00580
RKM00590
RKM00600
RKM00610
RKM00620
RKM00630
RKM00640
RKM00650
RKM00660
RKM00670
RKM00680
RKM00690
RKM00700

```

	WK(I,3)=H3*WK(I,1)	RKM00710
50	WK(I,6)=Y(I)+WK(I,3)	RKM00720
	CALL DIFEQ(N,XN+H3,WK(I,6),WK(I,2))	RKM00730
	DO 60 I=1,N	RKM00740
60	WK(I,6)=Y(I)+(WK(I,3)+H3*WK(I,2))/2.	RKM00750
	CALL DIFEQ(N,XN+H3,WK(I,6),WK(I,2))	RKM00760
	DO 70 I=1,N	RKM00770
	WK(I,4)=H3*WK(I,2)	RKM00780
70	WK(I,6)=Y(I)+(3.*WK(I,3)+9.*WK(I,4))/8.	RKM00790
	CALL DIFEQ(N,XN+H/2.,WK(I,6),WK(I,2))	RKM00800
	DO 80 I=1,N	RKM00810
	WK(I,5)=H3*WK(I,2)	RKM00820
80	WK(I,6)=Y(I)+(3.*WK(I,3)-9.*WK(I,4)+12.*WK(I,5))/2.	RKM00830
	CALL DIFEQ(N,XN+H,WK(I,6),WK(I,2))	RKM00840
CCCC	FIND THE LARGEST RELATIVE ERROR	RKM00850
	TEST=0.	RKM00860
	DO 90 I=1,N	RKM00870
	YX=Y(I)	RKM00880
	IF(YX.EQ. 0.) YX=ACCURC	RKM00890
	E=((WK(I,3)-9.*WK(I,4))/2.+4.*WK(I,5)-H3*WK(I,2))/2.)/YX	RKM00900
90	TEST=AMAX1(TEST,ABS(E))	RKM00910
	FIRST=.FALSE.	RKM00920
	IF(TEST.LT. ACCURC) GO TO 100	RKM00930
CCCC	IF THE LARGEST ERROR IS GREATER THAN ACCURC HALF THE STEP	RKM00940
	SIZE AND TRY AGAIN.	RKM00950
	H=H/2.	RKM00960
	IF(H.LT. HMIN) GO TO 10	RKM00970
	QUIT=.FALSE.	RKM00980
	GO TO 40	RKM00990
CCCC	TRUNCATION ERROR LESS THAN ACCURC, RESET THE Y ARRAY TO	RKM01000
	SET UP FOR THE NEXT INTERVAL	RKM01010
100	XN=XN+H	RKM01020
	DO 110 I=1,N	RKM01030
110	Y(I)=Y(I)+(WK(I,3)+4.*WK(I,5)+H3*WK(I,2))/2.	RKM01040
CCCC	CHECK FOR STEP SIZE DOUBLING. DOUBLE IF LARGEST RELATIVE	RKM01050
	ERROR IS 32 TIMES LESS THAN ACCURC.	RKM01060
	IF(.NOT.(TEST.GE. ACCURC/32. .OR. QUIT)) H=H*H	RKM01070
	IF(.NOT. QUIT) GO TO 20	RKM01080
	RETURN	RKM01090
CCCC	THE VALUE OF H (DEL) IS LESS THAN THE SPECIFIED MINIMUM.	RKM01100
	REPORT THIS AND ERROR OUT.	RKM01110
10	CONTINUE	RKM01120
1000	FORMAT('1 H BELOW HMIN'/ '0 INTEGRATION ABORTED')	RKM01130
	DEL=0.	RKM01140
	RETURN	RKM01150
	END	RKM01160
		RKM01170
		RKM01180
		RKM01190
		RKM01200
		RKM01210
		RKM01220
		RKM01230
		RKM01240
		RKM01250
		RKM01260

```

SUBROUTINE SETUP(NCHS,SRCBAS,SIDE1,SIDE2,TRNFRM)
DIMENSION NCHS(2),SRCBAS(2),SIDE1(2),SIDE2(2),TRNFRM(2,2),REF(2)
COMMON /ARRAY/OVRLAP,AREA,PERIM,PRJARY,CENDIF
COMMON/MOS/DIFF(2,200),NCHTOT,PRSEP(200),NTOT,NARY,ITOT,
+ COOR(2,200),TSTAG(200),DMMY(401)
COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLINT,KSTOR,NPLOTU
*****
C *****
C PURPOSE
C
C   TO CONVERT THE USER DEFINED COORDINATES OF THE CHARGES TO THE
C   LOCAL COORDINATE SYSTEM.
C
C INPUTS
C
C   NCHS - SINGLY DIMENSIONED ARRAY CONTAINING THE NUMBER OF
C           CHARGES.
C
C   SRCBAS - A REFERENCE CHARGE IN THE USER DEFINED COORDINATES.
C
C   SIDE1,SIDE2 - VECTORS DESCRIBING THE BOUNDING PARRALLELOGRAM.
C
C   TRNFRM - COORDINATE SYSTEM TRANSFORMATION MATRIX.
C
C OUTPUTS RETURNED IN COMMON /ARRAY/ AND /SEPRTN/
C
C   DIFF- DOUBLY DIMENSIONED ARRAY CONTAINING THE CHARGE COORDINATES
C           IN THE LOCAL COORDINATE SYSTEM.
C
C   ITOT- TOTAL NUMBER OF CHARGES.
C
C   NCHTOT- WHEN NARY=1 OR 2 THE TOTAL NUMBER OF CHARGES. WHEN
C            WHEN NARY=3 IS SET =1.
C
C   NTOT- WHEN NARY=1 OR 2 IS SET =1 AND WHEN NARY=3 IS THE TOTAL
C           NUMBER OF CHARGES.
C
C SUBROUTINES AND FUNCTIONS
C
C   UNIT COMPUTES THE UNIT VECTOR OF A GIVEN VECTOR
C *****
C *****
C   IF(NARY.NE.1)GO TO 4
C   NCHTOT=NCHS(1)*NCHS(2)
C   NTOT=1
C   ITOT=NCHTOT
C   GO TO 6
C 4 IF(NARY.NE.2)GO TO 5
C   NCHTOT=NCHS(1)
C   NTOT=1
C   ITOT=NCHTOT
C   GO TO 6
C 5 IF(NARY.NE.3)GO TO 998
C   NCHTOT=1
C   NTOT=NCHS(1)
C   ITOT=NTOT
C 6 CONTINUE
C
C DETERMINE THE COORDINATE OF THE REFERENCE CHARGE IN THE INTERNAL
C COORDINATE SYSTEM
C
C   DO 20 I=1,2
C   REF(1)=0.0
C   DO 10 J=1,2
C   REF(1)=REF(1)+TRNFRM(I,J)*SRCBAS(J)
C 10 CONTINUE
C 20 CONTINUE
C   DO 40 I=1,2
C   DO 30 J=1,ITOT

```

AD-A114 417 ARMY ELECTRONICS RESEARCH AND DEVELOPMENT COMMAND WS--ETC F/G 4/1
PROGRAM LISTINGS FOR EOSAEL 80-B AND ANCILLARY CODES AGAUS AND --ETC(11)
FEB 82 R G STEINHOFF
UNCLASSIFIED ERADCOM/ASL-TR-0107-V2-SU NL

3-6

3-6

3-6

3-6

3-6

3-6

3-6

3-6

3-6

3-6

3-6

3-6

3-6

3-6

3-6

3-6

3-6

3-6

3-6

3-6

3-6

3-6

3-6

3-6

3-6

3-6

3-6

3-6

3-6

3-6

3-6

3-6

3-6

3-6

3-6

	DIFF(I,J)=0.0	SET00710
30	CONTINUE	SET00720
40	CONTINUE	SET00730
	IF(NARY.GT.1)GO TO 90	SET00740
	NM=0	SET00750
	NC1=NCHS(1)	SET00760
	NC2=NCHS(2)	SET00770
C		SET00780
CCCC	COMPUTE LOCATIONS OF CHARGES FOR INTERNAL COORDINATE SYSTEM FOR	SET00790
	UNIFORMLY DISTRIBUTED CHARGES.	SET00800
	DO 80 M=1,NC2	SET00810
	DO 70 N=1,NC1	SET00820
	NM=NK+1	SET00830
	DO 60 I=1,2	SET00840
	DO 50 J=1,2	SET00850
	DIFF(I,NM)=DIFF(I,NM)+FLOAT(N-1)*TRNFRM(I,J)*SIDE1(J)	SET00860
	1 +FLOAT(M-1)*TRNFRM(I,J)*SIDE2(J)	SET00870
50	CONTINUE	SET00880
60	CONTINUE	SET00890
70	CONTINUE	SET00900
80	CONTINUE	SET00910
	GO TO 999	SET00920
90	CONTINUE	SET00930
C		SET00940
CCCC	TRANSFORM CHARGE LOCATIONS TO LOCAL COORDINATE SYSTEM FOR RANDOM	SET00950
	CHARGES.	SET00960
	NC1=NCHS(1)	SET00970
	DO 120 M=1,NC1	SET00980
	DO 110 I=1,2	SET00990
	DO 100 J=1,2	SET01000
	DIFF(I,M)=DIFF(I,M)+TRNFRM(I,J)*COORD(J,M)	SET01010
100	CONTINUE	SET01020
	DIFF(I,M)=DIFF(I,M)-REF(I)	SET01030
110	CONTINUE	SET01040
120	CONTINUE	SET01050
	GO TO 999	SET01060
998	WRITE(100UT,778)	SET01070
778	FORMAT(5X,23H *** NARY OUT OF RANGE)	SET01080
999	RETURN	SET01090
	END	SET01100
		SET01110
		SET01120

SUBROUTINE SOURCE(W,NCHRG,DD,NSOIL,DSOD)

PURPOSE

TO CALCULATE EXPLOSIVE DUST SOURCE TERM FOR THE
DIRTRAN CODE

INPUT

W THE WEIGHT OF THE CHARGE IN KG-TNT
DD DETONATION DEPTH IN METERS
NSOIL INTEGER SOIL INDEX
DSOD DEPTH OF SOD IN METERS

OUTPUT (RETURNED IN COMMON /PRTINF/, /BUOYCL/ AND /CARB/)

R0 INITIAL CLOUD RADIUS IN METERS
VGRAV SINGLE DIMENSIONED ARRAY CONTAINING OPTICALLY WEIGHTED
AVERAGE SETTLING VELOCITIES FOR EACH SIZE RANGE IN
THE PARTICLE DISTRIBUTION (METERS/SEC)
NPRTS THE NUMBER OF SIZE RANGES IN THE PARTITIONING OF THE
PARTICLE SIZE SPECTRUM
RSPH THE INITIAL RADIUS OF THE CLOUD IN METERS
DELT THE INITIAL DIFFERENCE IN TEMPERATURE BETWEEN THE CLOUD
AND SURROUNDINGS (DEGREES KELVIN)
VZSPH THE INITIAL VERTICAL VELOCITY OF THE CLOUD (M/S)
XCMSPH INITIAL HORIZONTAL POSITION OF THE CLOUD (METERS)
YCMSPH INITIAL Y POSITION OF THE CLOUD (METERS)
ZCMSPH INITIAL HEIGHT OF THE CLOUD (METERS)
XTOP INITIAL X POSITION OF THE TOP OF THE CLOUD (METERS)
YTOP INITIAL Y POSITION OF THE TOP OF THE CLOUD (METERS)
RISTIM TIME LAPSED SINCE DETONATION IN SECONDS

RCARB1 PORTION OF BUOYANT CLOUD WHICH IS DIRT PARTICLES

RCARB2 PORTION OF BUOYANT CLOUD WHICH IS CARBON PARTICLES

CALLED BY DUSTCL

SUBROUTINES AND FUNCTIONS

NONE

LOGICAL HORIZ,ONCE
DIMENSION CR(5,7),CD(5,7),OWML(3,4),OWSV(3,4),PRTTN(4)
DIMENSION S(3),BURHTR(5),WTRAT(5)
COMMON/PRTINF/,R0,VGRAV(3),NPRTS
COMMON/IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLINT,KSTOR,NPLOTUS
COMMON/BUOYCL/,RSPH,DELT,VZSPH,XCMSPH,YCMSPH,ZCMSPH,XTOP,YTOP,
* SPHNS(3),RISTIM
COMMON/EKTEMP/Z0,ZL,T0,TC1,TC2,TC3
COMMON/STARS/USTAR,TSTAR,ZSTAR
COMMON/WNDPRM/DXZ0,DYX0,DZ0,U0,UM,DN,ZINV
COMMON/BURST/ACCEL,TBURST
COMMON/GEOM/COSTH2,SINTH,SINTH2,VISEXT,RTPI,SCRN(2)
COMMON/MODE/HORIZ
COMMON/DISCS/NDSCS,TDSC(20),XDSC(20),ZDSC(20),R2DSC(20),
1 QDSC(20,3)
COMMON/CARB/RCARB1,RCARB2
COMMON/NTAL/TNOT,VOLSPH,TNO,CBLEED

CR IS THE CRATER RADIUS INDEXED BY COEFFICIENT AND SOIL TYPE

DATA CR/.271,-.684,.39,.886,0.,.271,-.684,.39,.886,0.,
1 .386,-.849,.367,.993,0.,.503,-.954,.45,1.19,0.,
2 .629,-1.08,.264,1.12,0.,.629,-1.08,.264,1.12,0.,

SORC0010
SORC0020
SORC0030
SORC0040
SORC0050
SORC0060
SORC0070
SORC0080
SORC0090
SORC0100
SORC0110
SORC0120
SORC0130
SORC0140
SORC0150
SORC0160
SORC0170
SORC0180
SORC0190
SORC0200
SORC0210
SORC0220
SORC0230
SORC0240
SORC0250
SORC0260
SORC0270
SORC0280
SORC0290
SORC0300
SORC0310
SORC0320
SORC0330
SORC0340
SORC0350
SORC0360
SORC0370
SORC0380
SORC0390
SORC0400
SORC0410
SORC0420
SORC0430
SORC0440
SORC0450
SORC0460
SORC0470
SORC0480
SORC0490
SORC0500
SORC0510
SORC0520
SORC0530
SORC0540
SORC0550
SORC0560
SORC0570
SORC0580
SORC0590
SORC0600
SORC0610
SORC0620
SORC0630
SORC0640
SORC0650
SORC0660
SORC0670
SORC0680
SORC0690
SORC0700

```

3 .806,-1.28,-.178,.852,0./
C CD IS THE CRATER DEPTH INDEXED BY COEFFICIENT AND SOIL TYPE
C DATA CD/.113,-.477,.27,1.84,1.05,.134,-.571,.343,2.24,1.31,
1 .189,-.84,.447,3.3,2.1,.251,-1.17,.494,4.72,3.34,
2 .189,-.84,.447,3.3,2.1,.331,-1.49,.579,4.92,3.13,
3 .449,-1.82,.322,4.11,2.02/
C OWML IS THE OPTICALLY WEIGHTED MASS LOADING COEFFICIENT INDEXED BY
C BIN SIZE AND SOIL TYPE
C DATA OWML/2.88E3,2*0.,3.08E3,8*0./
C OWSV IS THE OPTICALLY WEIGHTED PARTICLE SETTLING VELOCITY (CM/SEC)
C INDEXED BY BIN SIZE AND SOIL TYPE
C DATA OWSV/12*0./
C PRITN IS THE PARTITIONING RATIO INDEXED ON SOIL TYPE
C DATA PRITN/4*.8/
C BURHTR IS THE RATIO OF BURST HEIGHT TO INITIAL RADIUS AND WTRAT
C IS THE FRACTION OF THE TOTAL WEIGHT WHICH IS EFFECTIVE IN THE CLOUD
C DATA BURHTR/0.,4.,2.,4.,3./,WTRAT/.6,1.,.8,1.,.7/
C
C RISTM=0,
C XCMSPH=0,
C YCMSPH=0,
C XTOP=0,
C YTOP=0,
C TNO=T0,
C NPRTS=1
C SCARB IS THE OPTICALLY WEIGHTED CARBON PARTICLE LOADING COEFFICIENT
C SCARB=270.*W
C W3=(W*WTRAT(NCHRG))*3.333333
C R0=2.0*W3
C TAMB=T0+TMPCAL(Z0,ZL,R0)*TSTAR
C DELT=.57*TAMB
C RSPH=R0
C ZCMSPH=R0
C BURHT=BURHTR(NCHRG)*R0
C BURVZ=1.3*SQR(R0)
C TBURST=.15*R0
C VZSPH=2.*BURHT/TBURST-BURVZ
C ACCEL=(BURVZ-VZSPH)/TBURST
C VOLSPH=(4./3.)*3.141593*R0**3
C TNOT=T0+DELT
C CLAM=DD/W3
C CALCULATE CRATER RADIUS AND DEPTH
C
C ONCE=.FALSE.
C IF(NSOIL.EQ.1)IDX=4
C IF(NSOIL.EQ.2)IDX=3
C GO TO 70
60 IF(NSOIL.EQ.1)IDX=6
C IF(NSOIL.EQ.2)IDX=4
70 CONTINUE
C RC=CR(1,IDX)
C DC=CD(1,IDX)
C IF (CLAM.LT.1.E-30) GO TO 98
C TERM=1.
C DO 100 I=2,5
C TERM=TERM*CLAM
C RC=RC + CR(I,IDX)*TERM
C DC=DC + CD(I,IDX)*TERM

```

SORC0710
SORC0720
SORC0730
SORC0740
SORC0750
SORC0760
SORC0770
SORC0780
SORC0790
SORC0800
SORC0810
SORC0820
SORC0830
SORC0840
SORC0850
SORC0860
SORC0870
SORC0880
SORC0890
SORC0900
SORC0910
SORC0920
SORC0930
SORC0940
SORC0950
SORC0960
SORC0970
SORC0980
SORC0990
SORC1000
SORC1010
SORC1020
SORC1030
SORC1040
SORC1050
SORC1060
SORC1070
SORC1080
SORC1090
SORC1100
SORC1110
SORC1120
SORC1130
SORC1140
SORC1150
SORC1160
SORC1170
SORC1180
SORC1190
SORC1200
SORC1210
SORC1220
SORC1230
SORC1240
SORC1250
SORC1260
SORC1270
SORC1280
SORC1290
SORC1300
SORC1310
SORC1320
SORC1330
SORC1340
SORC1350
SORC1360
SORC1370
SORC1380
SORC1390
SORC1400

100	CONTINUE	SORC1410
98	CONTINUE	SORC1420
	RC=RC*W3	SORC1430
	DC=DC*(W*WTRAT(NCHRG))*3	SORC1440
C		SORC1450
CC	GET CRATER VOLUME	SORC1460
C		SORC1470
	DSDC=DSOD/DC	SORC1480
	VC=(2.*3.141592/3.)*RC*RC*DC*(1.-1.5*DSDC*(1.-DSDC*DSDC/3.))	SORC1490
	IF(DSOD.GE.DC)VC=0.0	SORC1500
	IF(ONCE)GO TO 110	SORC1510
	ONCE=.TRUE.	SORC1520
	IF(NSOIL.EQ.1)VC1=.5*VC	SORC1530
	IF(NSOIL.EQ.2)VC1=.25*VC	SORC1540
	GO TO 60	SORC1550
110	IF(NSOIL.EQ.1)VC=VC1+.5*VC	SORC1560
	IF(NSOIL.EQ.2)VC=VC1+.75*VC	SORC1570
C		SORC1580
CC	CALCULATE OPTICALLY WEIGHTED PARAMETERS	SORC1590
C		SORC1600
	NDSCS=MIN0(10,IFIX(5.*W3/1.8))	SORC1610
	CBLEED=0.	SORC1620
	DO 101 L=1,NPRTS	SORC1630
	S(L)=QWML(L,NSOIL)*VC	SORC1640
	VGRAY(L)=QWSV(L,NSOIL)	SORC1650
	SPHNS(L)=PRTTN(NSOIL)*S(L)	SORC1660
	QDSC(1,L)=(1.-PRTTN(NSOIL))*S(L)/FLOAT(NDSCS)	SORC1670
	CBLEED=CBLEED+S(L)	SORC1680
101	CONTINUE	SORC1690
	CBLEED=CBLEED*.03/W3**3	SORC1700
	RCARB=SCARB/SPHNS(1)	SORC1710
	RCARB1=1./(<1.+RCARB)	SORC1720
	RCARB2=RCARB/(<1.+RCARB)	SORC1730
	SPHNS(1)=SPHNS(1)+SCARB	SORC1740
	DELH=2.*R0/FLOAT(NDSCS)	SORC1750
	Z=-DELH/2.	SORC1760
	DO 200 I=1,NDSCS	SORC1770
	Z=Z+DELH	SORC1780
	ZDSC(I)=2	SORC1790
	DO 201 J=1,NPRTS	SORC1800
	QDSC(1,J)=QDSC(1,J)	SORC1810
201	CONTINUE	SORC1820
	CON=ALOG(QDSC(1,1)/VISEXT/DELH/(2.*R0)/3.14159)	SORC1830
	IF(CON.GT.1.)GO TO 210	SORC1840
	D=1.	SORC1850
	GO TO 230	SORC1860
210	D=CON	SORC1870
	DO 220 IT=1,5	SORC1880
	D=(CON-1.+ALOG(D))*D/(D-1)	SORC1890
220	CONTINUE	SORC1900
230	R2DSC(I)=4.*R0*R0/D	SORC1910
	TDSC(I)=-DELH*DELH/D/(DZ0*Z**DN)/4.	SORC1920
	SIGZ=DELH*DELH/D	SORC1930
	XDSC(I)=U0*Z**UM*TDSC(I)	SORC1940
200	CONTINUE	SORC1950
999	RETURN	SORC1960
	END	SORC1970

	SUBROUTINE TEMP(Z,TA,DTADZ)	TEMP0240
	PURPOSE	TEMP0010
	TO COMPUTE THE AMBIENT ATMOSPHERIC POTENTIAL TEMPERATURE AND	TEMP0020
	GRADIENT AT A GIVEN HEIGHT.	TEMP0030
	INPUTS	TEMP0040
	Z HEIGHT AT WHICH AMBIENT TEMPERATURE AND TEMPERATURE	TEMP0050
	GRADIENT ARE DESIRED.	TEMP0060
	OUTPUTS	TEMP0070
	TA AMBIENT POTENTIAL TEMPERATURE	TEMP0080
	DTADZ TEMPERATURE GRADIENT	TEMP0090
	SUBROUTINES AND FUNCTIONS NEEDED	TEMP0100
	TMPCAL COMPUTES SCALED TEMPERATURE AT A GIVEN HEIGHT	TEMP0110
	CALLED BY DIFEQ, ATMCAL	TEMP0120
	*****	TEMP0130
	COMMON/STARS/USTAR,TSTAR,ZSTAR	TEMP0140
	COMMON/EKWIND/ALP,C,PYF,PDF,UMAT,VHAT	TEMP0150
	COMMON/COEF/AW,CW,BW,DW,AT,CT,BT,DT	TEMP0160
	COMMON/EKTEMP/Z0,ZL,T0,TC1,TC2,TC3	TEMP0170
	COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLINT,KSTOR,NPLOTU	TEMP0180
	S=Z/ZL	TEMP0190
	TA=TSTAR*TMPCAL(Z0,ZL,Z)+T0	TEMP0200
	IF(ABS(ZL).LE.1.E03)GO TO 10	TEMP0210
	NEUTRAL CASE	TEMP0220
	DTADZ=TSTAR/(Z0+Z)	TEMP0230
	GO TO 999	TEMP0250
	10 IF(ZL.GT.0.0)GO TO 15	TEMP0260
	UNSTABLE CASE	TEMP0270
	DTADZ=(TSTAR/Z)*(1.-16.*(S)**(-1./2.))	TEMP0280
	IF(S.LT.-2.0)DTADZ=(TSTAR/ZL)*(AT/3.*(-ZL/Z)**(4./3.))	TEMP0290
	GO TO 999	TEMP0300
	STABLE CASE	TEMP0310
	15 DTADZ=(TSTAR/ZL)*(ZL/(Z0+Z)+11.)	TEMP0320
	IF(S.GT.1.5)DTADZ=BT*TSTAR/ZL	TEMP0330
	999 RETURN	TEMP0340
	END	TEMP0350
		TEMP0360
		TEMP0370
		TEMP0380
		TEMP0390
		TEMP0400
		TEMP0410
		TEMP0420
		TEMP0430
		TEMP0440
		TEMP0450
		TEMP0460
		TEMP0470
		TEMP0480
		TEMP0490
		TEMP0500
		TEMP0510

```

C      FUNCTION TMPCAL(Z0,ZL,Z)
C      *****
C      PURPOSE
C      TO CALCULATE THE POTENTIAL TEMPERATURE SCALED BY THE SCALE
C      TEMPERATURE, T*, FROM GIVEN FRICTION HEIGHT AND MONIN-OBUKHOV
C      LENGTH AT A SPECIFIED HEIGHT.
C      INPUTS
C      Z0      THE FRICTION HEIGHT IN METERS.
C      ZL      THE MONIN-OBUKHOV LENGTH IN METERS.
C      Z       THE HEIGHT AT WHICH THE SCALED VELOCITY IS DESIRED
C              IN METERS
C      RETURNS SCALED TEMPERATURE
C      CALLED BY ATMCAL, SOURCE AND TEMP
C      *****
C      LOGICAL LOW
C      COMMON/COEF/AW,CW,BW,DW,AT,CT,BT,DT
C      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
C      PHIM(Z)=(1.-16.*Z)**(-.25)
C      PSIH(S,S0)=ALOG(((S**2.-1.)/(S**2.+1.))*((S0**2.+1.)/(S0**2.-1.)))
C      PSHS(Z)=-11.*Z
C      TRACE 999
C      PHIM      THE SHEAR OF MOMENTUM
C      PSIH      THE UNIVERSAL FUNCTION FOR DEVIATION FROM LOGARITHMIC
C                POTENTIAL TEMPERATURE PROFILE IN THE BOUNDARY LAYER
C                OF AN UNSTABLE ATMOSPHERE
C      PHHS      THE SAME AS PHIH EXCEPT FOR STABLE ATMOSPHERE
C      IF(ABS(ZL).LE.1.E3)GO TO 100
C      TMPCAL=ALOG(1.+Z/Z0)
C      GO TO 999
100  CONTINUE
C      P=SIGN(1.,ZL)
C      LOW=.TRUE.
C      S=Z/ZL
C      IF(S.LE.1.5.AND.S.GE.-2.)GO TO 10
C      S=AMIN1(S,1.5)
C      S=AMAX1(S,-2.)
C      LOW=.FALSE.
10   CONTINUE
C      IF(P)120,130,130
120  S=1./PHIM(S)
C      S1=Z0/ZL
C      S0=1./PHIM(S1)
C      TMPCAL=PSIH(S,S0)
C      FIND CONSTANTS FOR MATCHING IN UNSTABLE CASE AT Z/ZL=-2.
C      S2=-2.
C      AT=-3.*(1.-16.*S2)**(-1./2.)*(-S2)**(1./3.)
C      CT=-1.*AT*(-S2)**(-1./3.)
C      GO TO 52
130  CONTINUE
C      PSI=PSHS(S)
C      TMPCAL=ALOG(1.+S*ZL/Z0)-PSI
C      FIND CONSTANTS FOR MATCHING IN STABLE CASE AT Z/ZL=1.5

```

```

TMP00010
TMP00020
TMP00030
TMP00040
TMP00050
TMP00060
TMP00070
TMP00080
TMP00090
TMP00100
TMP00110
TMP00120
TMP00130
TMP00140
TMP00150
TMP00160
TMP00170
TMP00180
TMP00190
TMP00200
TMP00210
TMP00220
TMP00230
TMP00240
TMP00250
TMP00260
TMP00270
TMP00280
TMP00290
TMP00300
TMP00310
TMP00320
TMP00330
TMP00340
TMP00350
TMP00360
TMP00370
TMP00380
TMP00390
TMP00400
TMP00410
TMP00420
TMP00430
TMP00440
TMP00450
TMP00460
TMP00470
TMP00480
TMP00490
TMP00500
TMP00510
TMP00520
TMP00530
TMP00540
TMP00550
TMP00560
TMP00570
TMP00580
TMP00590
TMP00600
TMP00610
TMP00620
TMP00630
TMP00640
TMP00650
TMP00660
TMP00670
TMP00680
TMP00690
TMP00700

```

```

C
S2=1.5
BT=1./(<Z0/ZL+S2>)+11.
DT=-1.*BT*S2
52 CONTINUE
IF(<LOW>)GO TO 999
IF(<P>)53,53,54
53 TMPCAL=TMPCAL+CT+AT*(<-ZL/Z>)*(1./3.)
GO TO 999
54 TMPCAL=TMPCAL+DT+BT*Z/ZL
999 RETURN
END

```

```

TMP00710
TMP00720
TMP00730
TMP00740
TMP00750
TMP00760
TMP00770
TMP00780
TMP00790
TMP00800
TMP00810
TMP00820

```

SUBROUTINE TRAP(TRSK,TRWK,TRSP,H,SIGW,SIG0,TIME,SKT,WAK,SPH)
THIS SUBROUTINE PERFORMS A TRAPEZOID INTEGRATION
INPUTS

TRSK - ESTIMATE TO THE CLOSEST POINT ALONG THE LINE OF SIGHT
TO THE CENTER OF THE SKIRT
TRWK - ESTIMATE TO THE CLOSEST POINT ALONG THE LINE OF SIGHT
TO THE CENTER OF THE WAKE
TRSP - ESTIMATE TO THE CLOSEST POINT ALONG THE LINE OF SIGHT
TO THE CENTER OF THE SPHERE
H - INTEGRATION STEP SIZE THROUGH THE SKIRT
SIGW - INTEGRATION STEP SIZE THROUGH THE WAKE
SIG0 - INTEGRATION STEP SIZE THROUGH THE SPHERE
TIME - TIME TRANSMITTANCE IS DESIRED

ALL OTHER NEEDED INFORMATION IS PASSED VIA COMMON BLOCKS

OUTPUTS

SKT - CONTRIBUTION TO THE CONCENTRATION ALONG THE LINE OF
SIGHT FROM THE SKIRT
WAK - CONTRIBUTION TO THE CONCENTRATION ALONG THE LINE OF
SIGHT FROM THE WAKE ONCE THE BUOYANT SPHERE HAS
CONVERTED TO THE WIND MODEL
SPH - CONTRIBUTION TO THE CONCENTRATION ALONG THE LINE OF
SIGHT FROM THE SPHERE ONCE IT HAS CONVERTED TO THE
WIND MODEL

FUNCTIONS NEEDED

CWIND - USED TO FIND THE CONCENTRATION AT A SPECIFIED POINT
(X,Y,Z) ALONG THE LINE OF SIGHT DUE TO THE SKIRT
CWAKE - USED TO FIND THE CONCENTRATION AT A SPECIFIED POINT
(X,Y,Z) ALONG THE LINE OF SIGHT DUE TO THE WAKE
CSPHER - USED TO FIND THE CONCENTRATION AT A SPECIFIED POINT
(X,Y,Z) ALONG THE LINE OF SIGHT DUE TO THE SPHERE

REAL KX,KZ
LOGICAL SWITCH,CHANGE,ONCE
DIMENSION OWF(5,2),OWFC(5),TRSK(3),TRWK(3),TRSP(3)
COMMON/LOS/TR(3),RE(3),U(3)
COMMON/ACL/CWINDS,CWINDC,CWINDW
COMMON/CARB/RCARB1,RCARB2
COMMON/TRAN/VTR,KZ,KX,TTR,XTR,ZTR,QPUFF(3),SWITCH,CHANGE
COMMON/SIG/SIG02,SIGC
COMMON/POINTS/XNORM,DOT1,DOT2,DOT3
DATA OWF/1.,.93,.52,.44,2.E-03,1.,1.,1.,1.,4.E-03/
DATA OWFC/1.,.95,.5,.2,1.E-03/
TI=TIME
SKT=0.0
WAK=0.0
SPH=0.0
SUM=0.0
SUM1=0.0
SUM2=0.0
IND=20

TRP00510
TRP00010
TRP00020
TRP00030
TRP00040
TRP00050
TRP00060
TRP00070
TRP00080
TRP00090
TRP00100
TRP00110
TRP00120
TRP00130
TRP00140
TRP00150
TRP00160
TRP00170
TRP00180
TRP00190
TRP00200
TRP00210
TRP00220
TRP00230
TRP00240
TRP00250
TRP00260
TRP00270
TRP00280
TRP00290
TRP00300
TRP00310
TRP00320
TRP00330
TRP00340
TRP00350
TRP00360
TRP00370
TRP00380
TRP00390
TRP00400
TRP00410
TRP00420
TRP00430
TRP00440
TRP00450
TRP00460
TRP00470
TRP00480
TRP00490
TRP00500
TRP00520
TRP00530
TRP00540
TRP00550
TRP00560
TRP00570
TRP00580
TRP00590
TRP00600
TRP00610
TRP00620
TRP00630
TRP00640
TRP00650
TRP00660
TRP00670
TRP00680
TRP00690
TRP00700

	ONCE=.FALSE.	TRP00710
	FP=0.0	TRP00720
	F=0.0	TRP00730
	J=0	TRP00740
	DO 10 I=1,IND	TRP00750
	XI=H*FLOAT(I-1)	TRP00760
	DIST=DOT1+XI	TRP00770
	IF(DIST.GT.XNORM)GO TO 10	TRP00780
	IF(DIST.LT.0.0)GO TO 10	TRP00790
	X=TRSK(1)+XI*U(1)	TRP00800
	Y=TRSK(2)+XI*U(2)	TRP00810
	Z=TRSK(3)+XI*U(3)	TRP00820
	FP=F	TRP00830
	ACL=CWIND(X,Y,Z,TI)	TRP00840
	F=ACL	TRP00850
C	CHECK TO SEE IF THE CONTRIBUTION IS NEGLIGABLE	TRP00860
C	IF(.NOT.ONCE)ACL1=ACL	TRP00870
C	IF(ACL1.LT.1.E-10)GO TO 11	TRP00880
	PER=.01*ACL1	TRP00890
	ONCE=.TRUE.	TRP00900
	J=J+1	TRP00910
	IF(ACL.LT.PER)GO TO 11	TRP00920
	IF(J.LE.1)GO TO 10	TRP00930
	SUM=SUM+FP+F	TRP00940
10	CONTINUE	TRP00950
11	CONTINUE	TRP00960
	IF((TIME-TTR).LT.1.E-20)GO TO 31	TRP00970
C	COMPUTE THE CONTRIBUTION TO CL FROM THE SPHERE USING A TRAPEZOID	TRP00980
C	INTEGRATION ONLY AFTER THE BUOYANT FIREBALL HAS CONVERTED TO THE WIND	TRP00990
C	MODEL	TRP01000
	ONCE=.FALSE.	TRP01010
	FP=0.0	TRP01020
	F=0.0	TRP01030
	J=0	TRP01040
	DO 20 I=1,IND	TRP01050
	XI=SIG0*FLOAT(I-1)	TRP01060
	DIST=DOT3+XI	TRP01070
	IF(DIST.GT.XNORM)GO TO 20	TRP01080
	IF(DIST.LT.0.0)GO TO 20	TRP01090
	X=TRSP(1)+XI*U(1)	TRP01100
	Y=TRSP(2)+XI*U(2)	TRP01110
	Z=TRSP(3)+XI*U(3)	TRP01120
	FP=F	TRP01130
	ACL=CSFER(X,Y,Z,TI)	TRP01140
	F=ACL	TRP01150
	IF(.NOT.ONCE)ACL1=ACL	TRP01160
	IF(ACL1.LT.1.E-05)GO TO 21	TRP01170
	PER=.01*ACL1	TRP01180
	ONCE=.TRUE.	TRP01190
	J=J+1	TRP01200
	IF(ACL.LT.PER)GO TO 21	TRP01210
	IF(J.LE.1)GO TO 20	TRP01220
	SUM1=SUM1+FP+F	TRP01230
20	CONTINUE	TRP01240
21	CONTINUE	TRP01250
C	COMPUTE CONTRIBUTION TO CL FROM THE WAKE AFTER THE BUOYANT FIREBALL	TRP01260
C	HAS CONVERTED TO THE WIND MODEL USING TRAPEZOID INTEGRATION WITH STEP	TRP01270
C	SIZE SIGW.	TRP01280
	ONCE=.FALSE.	TRP01290
	FP=0.0	TRP01300
	F=0.0	TRP01310
	J=0	TRP01320
	DO 30 I=1,IND	TRP01330
	XI=SIGW*FLOAT(I-1)	TRP01340
		TRP01350
		TRP01360
		TRP01370
		TRP01380
		TRP01390
		TRP01400


```

DIST=DOT2+XI
IF(DIST.GT.XNORM)GO TO 30
IF(DIST.LT.0.0)GO TO 30
X=TRWK(1)+XI*U(1)
Y=TRWK(2)+XI*U(2)
Z=TRWK(3)+XI*U(3)
FP=F
ACL=CWAKE(X,Y,Z,TI)
F=ACL
IF(.NOT.ONCE)ACL1=ACL
IF(ACL1.LT.1.E-05)GO TO 31
PER=.01*ACL1
ONCE=.TRUE.
J=J+1
IF(ACL.LT.PER)GO TO 31
IF(J.LE.1)GO TO 30
SUM2=SUM2+FP+F
30 CONTINUE
31 CONTINUE
SKT=(ABS(H)/2.)*SUM
WAK=(ABS(SIGW)/2.)*SUM2
SPH=(ABS(SIGO)/2.)*SUM1
999 RETURN
END

```

```

TRP001410
TRP001420
TRP001430
TRP001440
TRP001450
TRP001460
TRP001470
TRP001480
TRP001490
TRP001500
TRP001510
TRP001520
TRP001530
TRP001540
TRP001550
TRP001560
TRP001570
TRP001580
TRP001590
TRP001600
TRP001610
TRP001620
TRP001630
TRP001640

```

```

SUBROUTINE TRNCAL(TRN,REC,TIME,TRNLOS)
CONTROLLING ROUTINE FOR CALCULATING TRANSMITTANCES FOR CHARGE
DISTRIBUTION TYPES 1 AND 2
INPUTS
TRN - TRANSMITTER COORDINATES IN THE LOCAL COORDINATE SYSTEM
REC - RECEIVER COORDINATES IN THE LOCAL COORDINATE SYSTEM
TIME - TIME AFTER THE DETONATION AT WHICH A TRANSMITTANCE IS DESIRED
ALL OTHER NECESSARY INPUTS ARE PASSED IN COMMON BLOCKS
OUTPUT
TRNLOS - TRANSMITTANCE ALONG THE SPECIFIED LINE OF SIGHT
SUBROUTINES NEEDED
AVRG - FINDS THE AVERAGE OF THE MOMENTS FOR THE DISCS
VSUM - ADDS TWO VECTORS
UNIT - DETERMINE A UNIT VECTOR
TRNCAL- DETERMINE THE LENGTH OF THE INTERSECTION OF THE LINE OF
SIGHT WITH THE WAKE AND SPHERE
TRAP - DOES A TRAPAZOIDAL INTEGRATION THROUGH SKIRT WAKE AND S
SPHERE FOR NON-HORIZONTAL LINES OF SIGHT
TRNCHK- CHECKS TO SEE IF THE OBSCURATION IS SUCH THAT THE
TRANSMITTANCE IS LESS THAN A SPECIFIED VALUE
FUNCTIONS NEEDED
DOTPRD - FINDS THE DOTPRODUCT OF TWO VECTORS
CWIND - FINDS THE CONCENTRATION ALONG A SPECIFIED HORIZONTAL
LINE OF SIGHT OR DETERMINES THE CONCENTRATION AT SOME
POINT ALONG THE LINE OF SIGHT FROM THE SKIRT
CWAKE - SAME AS CWIND EXCEPT FOR WAKE
CSPHER - SAME AS CWIND EXCEPT FOR BUOYANT SPHERE
*****
REAL KZ,KX
DIMENSION TRN(3),REC(3),OWF(5,2),OWFC(5),TEMP(2)
DIMENSION DIR(2), XW(3),XS(3)
DIMENSION TRSK(3),TRWK(3),TRSP(3)
LOGICAL HORIZ,SWITCH,CHANGE,TEST,SKIP
COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUT
COMMON/CARB/RCARB1,RCARB2
COMMON/BUOYCL/RSPH,DELT,VZ,XCM,YCM,ZCM,XTOP,YTOP,SPHNS(3),RISTIM
COMMON/M05/DIFF(2,200),NCHTOT,PRSEP(200),NTOT,NARY,ITOT,
+ DMM(600),DMMY(401)
COMMON/GEOM/COSTH2,SINTH,SINTH2,VISEXT,RTPI,SCRN(2)
COMMON/MODE/HORIZ
COMMON/TRAN/VTR,KZ,KX,TTR,XTR,ZTR,QPUFF(3),SWITCH,CHANGE
COMMON/ACL/CWINDS,CWINDC,CWINDW
COMMON/LOS/T(3),R(3),U(3)
COMMON/SIG/SIG02,SIGC
COMMON/CHARGE/NCHG
COMMON/POINTS/XNORM,DOT1,DOT2,DOT3

```

```

TRL00520
TRL00010
TRL00020
TRL00030
TRL00040
TRL00050
TRL00060
TRL00070
TRL00080
TRL00090
TRL00100
TRL00110
TRL00120
TRL00130
TRL00140
TRL00150
TRL00160
TRL00170
TRL00180
TRL00190
TRL00200
TRL00210
TRL00220
TRL00230
TRL00240
TRL00250
TRL00260
TRL00270
TRL00280
TRL00290
TRL00300
TRL00310
TRL00320
TRL00330
TRL00340
TRL00350
TRL00360
TRL00370
TRL00380
TRL00390
TRL00400
TRL00410
TRL00420
TRL00430
TRL00440
TRL00450
TRL00460
TRL00470
TRL00480
TRL00490
TRL00500
TRL00510
TRL00530
TRL00540
TRL00550
TRL00560
TRL00570
TRL00590
TRL00600
TRL00610
TRL00620
TRL00630
TRL00640
TRL00650
TRL00660
TRL00670
TRL00680
TRL00690
TRL00700

```

	COMMON/TRANNY/THRESH,TEST,NWL,NSOIL	TRL00710
	COMMON/SKIPIT/SKIP	TRL00720
	COMMON/EKTEMP/20,ZL,T0,TC1,TC2,TC3	TRL00730
	DATA ONEM/-1.0/	TRL00740
	DATA QWF/1.,.93,.52,.44,2.E-03,1.,1.,1.,1.,4.E-03/	TRL00750
	DATA QWFC/1.,.95,.5,.2,1.E-03/	TRL00760
C	PARAMETERIZE THE LINE CONNECTING THE TRANSMITTER AND RECEIVER	TRL00770
C	NCHG=1	TRL00780
C	SKIP=.FALSE.	TRL00790
	TEST=.FALSE.	TRL00800
	HSPH=0.0	TRL00810
	HWAK=0.0	TRL00820
	ACLSKT=0.0	TRL00830
	ACLWAK=0.0	TRL00840
	ACLSPH=0.0	TRL00850
	XNORM=0.0	TRL00860
	DO 10 I=1,3	TRL00870
	R(I)=REC(I)	TRL00880
	T(I)=TRN(I)	TRL00890
	U(I)=R(I)-T(I)	TRL00900
	XNORM=XNORM+U(I)**2	TRL00910
10	CONTINUE	TRL00920
	XNORM=SQRT(XNORM)	TRL00930
	U(1)=U(1)/XNORM	TRL00940
	U(2)=U(2)/XNORM	TRL00950
	U(3)=U(3)/XNORM	TRL00960
	IF((TIME-TTR).LT.1.E-20)GO TO 14	TRL00970
C	IF THE BUOYANT SPHERE HAS BECOME WIND BLOWN DETERMINE THE CENTER OF	TRL00980
C	MASS OF THE REFERENCE CHARGE.	TRL00990
C	XCM=XTR+VTR*(TIME-TTR)	TRL01000
	ZCM=ZTR	TRL01010
C	DETERMINE CENTER OF WAKE FOR REFERENCE CHARGE	TRL01020
C	14 ZX=5.0	TRL01030
	CALL AVRG(ZX,TIME,QTOT,XBAVRG,SIG2X,SIG2Y)	TRL01040
	XW(1)=(XBAVRG+XCM)/2.	TRL01050
	XW(2)=YCM/2.	TRL01060
	XW(3)=(5.0+ZCM)/2.	TRL01070
C	IF THE DIFFERENCE BETWEEN THE TRANSMITTER AND RECEIVER IS GREATER	TRL01080
C	THAN 1 PERCENT OF THE DISTANCE BETWEEN THEM THEN THE LOS IS	TRL01090
C	CONSIDERED A SLANT PATH.	TRL01100
C	13 IF(ABS(U(3)).GT..01)GO TO 20	TRL01110
	IF(ABS(TRN(3)-REC(3)).LT.1.E-06)GO TO 9	TRL01120
C	COMPUTE CLOSEST POINT ALONG THE LOS TO OUR ESTIMATE OF THE CENTER OF	TRL01130
C	THE WAKE	TRL01140
	DOT=-(U(1)*(T(1)-XW(1))+U(2)*(T(2)-XW(2))+U(3)*(T(3)-XW(3)))	TRL01150
	T(3)=T(3)+U(3)*DOT	TRL01160
	R(3)=T(3)	TRL01170
C	COMPUTE CONTRIBUTIONS FOR A HORIZONTAL PATH	TRL01180
C	9 HORIZ=.TRUE.	TRL01190
	CALL VSUM(REC,TRN,ONEM,DIR)	TRL01200
	CALL UNIT(DIR,DIR,RANGE)	TRL01210
	COSTH=DIR(1)	TRL01220
	SINTH=DIR(2)	TRL01230
	SINTH2=SINTH*SINTH	TRL01240
	COSTH2=COSTH**2	TRL01250
	SCRN(1)=SINTH	TRL01260
	SCRN(2)=-COSTH	TRL01270
	DO 12 J=1,ITOT	TRL01280
		TRL01290
		TRL01300
		TRL01310
		TRL01320
		TRL01330
		TRL01340
		TRL01350
		TRL01360
		TRL01370
		TRL01380
		TRL01390
		TRL01400

```

DO 11 I=1,2                                TRL01410
TEMP(I)=DIFF(I,J)                          TRL01420
11 CONTINUE                                TRL01430
PRSEP(J)=DOTPRD(TEMP,SCRN)                 TRL01440
12 CONTINUE                                TRL01450
X=DOTPRD(SCRN,TRN)                         TRL01460
C                                           TRL01470
C COMPUTE THE CONTRIBUTION FROM THE SKIRT AT A HEIGHT OF T(3) WHERE T(3) IS THE HEIGHT OF THE TRANSMITTER IF THE DIFFERENCE BETWEEN THE TRANSMITTER AND RECEIVER HEIGHTS IS SMALL AND IS THE Z COMPONENT OF THE POINT ON THE LINE CONNECTING THE TRANSMITTER AND RECEIVER WHICH IS CLOSEST TO OUR ESTIMATE OF THE CENTER OF THE WAKE OTHERWISE. TRL01480
C                                           TRL01490
C                                           TRL01500
C                                           TRL01510
C                                           TRL01520
C                                           TRL01530
C                                           TRL01540
C                                           TRL01550
C                                           TRL01560
C                                           TRL01570
C                                           TRL01580
C                                           TRL01590
C                                           TRL01600
C                                           TRL01610
C                                           TRL01620
C                                           TRL01630
C                                           TRL01640
C                                           TRL01650
C                                           TRL01660
C                                           TRL01670
C                                           TRL01680
C                                           TRL01690
C                                           TRL01700
C                                           TRL01710
C                                           TRL01720
C                                           TRL01730
C                                           TRL01740
C                                           TRL01750
C                                           TRL01760
C                                           TRL01770
C                                           TRL01780
C                                           TRL01790
C                                           TRL01800
C                                           TRL01810
C                                           TRL01820
C                                           TRL01830
C                                           TRL01840
C                                           TRL01850
C                                           TRL01860
C                                           TRL01870
C                                           TRL01880
C                                           TRL01890
C                                           TRL01900
C                                           TRL01910
C                                           TRL01920
C                                           TRL01930
C                                           TRL01940
C                                           TRL01950
C                                           TRL01960
C                                           TRL01970
C                                           TRL01980
C                                           TRL01990
C                                           TRL02000
C                                           TRL02010
C                                           TRL02020
C                                           TRL02030
C                                           TRL02040
C                                           TRL02050
C                                           TRL02060
C                                           TRL02070
C                                           TRL02080
C                                           TRL02090
C                                           TRL02100

DO 11 I=1,2                                TRL01410
TEMP(I)=DIFF(I,J)                          TRL01420
11 CONTINUE                                TRL01430
PRSEP(J)=DOTPRD(TEMP,SCRN)                 TRL01440
12 CONTINUE                                TRL01450
X=DOTPRD(SCRN,TRN)                         TRL01460

C COMPUTE THE CONTRIBUTION FROM THE SKIRT AT A HEIGHT OF T(3) WHERE T(3) IS THE HEIGHT OF THE TRANSMITTER IF THE DIFFERENCE BETWEEN THE TRANSMITTER AND RECEIVER HEIGHTS IS SMALL AND IS THE Z COMPONENT OF THE POINT ON THE LINE CONNECTING THE TRANSMITTER AND RECEIVER WHICH IS CLOSEST TO OUR ESTIMATE OF THE CENTER OF THE WAKE OTHERWISE.

ACLSKT=CWIND(X,Y,T(3),TIME)
IF(TEST)GO TO 998

TEST IS A LOGICAL VARIABLE RETURNED IN COMMON/TRANNY/ FROM SUBROUTINE TRNCHK WHICH IS CALLED BY CWIND, CWAKE, CSPHER, TRNCLD, AND TRAP EACH TIME A CONTRIBUTION IS MADE TO THE OPTICALLY WEIGHTED CONCENTRATION ALONG THE OPTICAL PATH

TEST=.FALSE. TRANSMITTANCE IS GREATER THAN TRNMIN
              (A TRANSMITTANCE THRESHOLD)
              =.TRUE. TRANSMITTANCE IS LESS THAN TRNMIN

IF(TIME.GT.TTR)ACLWAK=CWAKE(X,Y,T(3),TIME)
IF(TEST)GO TO 998
IF(TIME.GT.TTR)ACLSPH=CSPHER(X,Y,T(3),TIME)
IF(TEST)GO TO 998
IF(TIME.GT.TTR)GO TO 50
CWINDS=ACLWAK
CWINDW=ACLWAK
CWINDC=ACLSPH
CALL TRNCLD(XNORM,TIME,ACLWAK,ACLSPH)
IF(TEST)GO TO 998
GO TO 50

DO TRAPEZOIDAL INTEGRATION FOR SLANT PATH IN BOTH DIRECTIONS FROM AN ESTIMATE OF THE LOCATION OF CENTER OF THE SKIRT USING A STEP SIZE OF SIG, THE GEOMETRIC MEAN OF THE AVERAGE OF THE SPREADS OF THE DISCS IN BOTH THE X AND Y DIRECTION, THEN IF THE BUOYANT SPHERE HAS CONVERTED TO THE WIND MODEL DO THE SAME FOR THE WAKE AND SPHERE WITH THE APPROPRIATE STEP SIZE. (CHECK TO SEE IF SPHERE HAS CONVERTED TO THE WIND MODEL IS DONE IN TRAP.)

20 HORIC=.FALSE.
SIGX=SQRT(SIG2X)
SIGY=SQRT(SIG2Y)
SIG=SQRT(SIGX*SIGY)
CALL WIND(2.0,UX,UY)
XS(1)=TIME*UX
XS(2)=TIME*UY
XS(3)=2.0

FIND THE POINTS ON THE LINE CONNECTING THE TRANSMITTER AND RECEIVER THAT ARE CLOSEST TO OUR ESTIMATE OF THE CENTER OF THE SKIRT, WAKE, AND SPHERE.

DO 48 J=1,ITOT
XSK=XS(1)+DIFF(1,J)
YSK=XS(2)+DIFF(2,J)
ZSK=XS(3)
DOT1=-(U(1)*(T(1)-XSK)+U(2)*(T(2)-YSK)+U(3)*(T(3)-ZSK))
IF(DOT1.LT.0.0)DOT1=0.0
IF(DOT1.GT.XNORM)DOT1=XNORM
XWK=XWK(1)+DIFF(1,J)
YWK=XWK(2)+DIFF(2,J)
ZWK=XWK(3)
DOT2=-(U(1)*(T(1)-XWK)+U(2)*(T(2)-YWK)+U(3)*(T(3)-ZWK))
IF(DOT2.LT.0.0)DOT2=0.0

```

IF(DOT2.GT.XNORM)DOT2=XNORM	TRL 02110
XSP=XCM+DIFF(1,J)	TRL 02120
YSP=YCM+DIFF(2,J)	TRL 02130
ZSP=ZCM	TRL 02140
DOT3=-(U(1)*(T(1)-XSP)+U(2)*(T(2)-YSP)+U(3)*(T(3)-ZSP))	TRL 02150
IF(DOT3.LT.0.0)DOT3=0.0	TRL 02160
IF(DOT3.GT.XNORM)DOT3=XNORM	TRL 02170
NCHG=J	TRL 02180
DO 45 II=1,3	TRL 02190
TRSK(II)=T(II)+U(II)*DOT1	TRL 02200
TRWK(II)=T(II)+U(II)*DOT2	TRL 02210
TRSP(II)=T(II)+U(II)*DOT3	TRL 02220
45 CONTINUE	TRL 02230
DIFS=DIFFUS(Z0,ZL,2.0)	TRL 02240
SIG2=SQRT(2.*DIFS*TIME)	TRL 02250
IF(SIG2.LT.1.0)SIG2=1.0	TRL 02260
H=U(1)*SIGX+U(2)*SIGY+U(3)*SIGZ	TRL 02270
IF(TIME.LT.TTR)GO TO 46	TRL 02280
HSPX=SQRT(SIG2+2.*KX*(TIME-TTR))	TRL 02290
HSPY=HSPX	TRL 02300
HSPZ=SQRT(SIG2+2.*KZ*(TIME-TTR))	TRL 02310
HSPH=U(1)*HSPX+U(2)*HSPY+U(3)*HSPZ)/2.	TRL 02320
HWAK=(HSPH+H)/2.	TRL 02330
46 CONTINUE	TRL 02340
CALL TRAP<TRSK,TRWK,TRSP,H,HWAK,HSPH,TIME,SKT,WAK,SPH>	TRL 02350
ACLSKT=ACLSKT+SKT	TRL 02360
ACLWAK=ACLWAK+WAK	TRL 02370
ACLSPH=ACLSPH+SPH	TRL 02380
CALL TRNCHK<ACLSKT,ACLWAK,ACLSPH>	TRL 02390
IF<TEST>GO TO 998	TRL 02400
H=-H	TRL 02410
IF(TIME.LT.TTR)GO TO 47	TRL 02420
HSPH=-HSPH	TRL 02430
HWAK=-HWAK	TRL 02440
47 CONTINUE	TRL 02450
CALL TRAP<TRSK,TRWK,TRSP,H,HWAK,HSPH,TIME,SKT,WAK,SPH>	TRL 02460
ACLSKT=ACLSKT+SKT	TRL 02470
ACLWAK=ACLWAK+WAK	TRL 02480
ACLSPH=ACLSPH+SPH	TRL 02490
CWINDS=ACLSKT	TRL 02500
CWINDW=ACLWAK	TRL 02510
CWINDS=ACLSPH	TRL 02520
CALL TRNCHK<ACLSKT,ACLWAK,ACLSPH>	TRL 02530
IF<TEST>GO TO 998	TRL 02540
48 CONTINUE	TRL 02550
IF(TIME.GT.TTR)GO TO 50	TRL 02560
CALL TRNCLD<XNORM,TIME,ACLWAK,ACLSPH>	TRL 02570
IF<TEST>GO TO 998	TRL 02580
50 CONTINUE	TRL 02590
ACLC=(ACLWAK+ACLSPH)*(RCARB1*OWF(NWL,NSOIL)+RCARB2*OWF(NWL))	TRL 02600
ACLS=ACLSKT*OWF(NWL,NSOIL)	TRL 02610
COMPUTE THE TRANSMITTANCE ALONG THE LINE OF SIGHT	TRL 02620
TRNLOS=EXP(-ACLS-ACLC)	TRL 02630
GO TO 999	TRL 02640
998 TRNLOS=0.0	TRL 02650
999 RETURN	TRL 02660
END	TRL 02670
	TRL 02680
	TRL 02690

```

SUBROUTINE TRNCHK(ACLS,ACLW,ACLC)
THIS IS A SUBROUTINE TO CHECK IF THE CONCENTRATIONS ARE HIGH ENOUGH
SO THAT THE TRANSMITTANCE WILL BE BELOW A GIVEN LEVEL.
INPUTS
  ACLS - CONTRIBUTION TO THE CONCENTRATION ALONG THE LINE OF SIGHT
        DUE TO THE SKIRT
  ACLW - CONTRIBUTION TO THE CONCENTRATION ALONG THE LINE OF SIGHT
        DUE TO THE WAKE
  ACLC - CONTRIBUTION TO THE CONCENTRATION ALONG THE LINE OF SIGHT
        DUE TO THE BUOYANT SPHERE
  ALL OTHER NECESSARY INFORMATION IS PASSED VIA COMMON BLOCKS
OUTPUT
  TEST - LOGICAL VARIABLE PASSED IN COMMON/TRANNY/ THAT IS .TRUE.
        IF THE CONCENTRATION IS SUCH THAT THE TRANSMITTANCE
        ALONG THE LINE OF SIGHT WILL BE LESS THAN THE THRESHOLD
        VALUE AND IS .FALSE. OTHERWISE
FUNCTIONS ANT SUBROUTINE NEEDED
  NONE
*****
  DIMENSION OWF(5,2),OWFC(5)
  LOGICAL TEST
  COMMON/CARB/RCARB1,RCARB2
  COMMON/TRANNY/THRESH,TEST,NWL,NSOIL
  DATA OWF/1.,.93,.52,.44,2.E-03,1.,1.,1.,1.,4.E-03/
  DATA OWFC/1.,.95,.5,.2,1.E-03/
  TEST=.FALSE.
  ACL=(ACLW+ACLC)*(RCARB1*OWF(NWL,NSOIL)+RCARB2*OWFC(NWL))
  ACL=ACL+ACLS*OWF(NWL,NSOIL)
  IF(ACL.GT.THRESH)TEST=.TRUE.
999 RETURN
END

```

```

TRK00310
TRK00010
TRK00020
TRK00030
TRK00040
TRK00050
TRK00060
TRK00070
TRK00080
TRK00090
TRK00100
TRK00110
TRK00120
TRK00130
TRK00140
TRK00150
TRK00160
TRK00170
TRK00180
TRK00190
TRK00200
TRK00210
TRK00220
TRK00230
TRK00240
TRK00250
TRK00260
TRK00270
TRK00280
TRK00290
TRK00300
TRK00320
TRK00330
TRK00340
TRK00350
TRK00360
TRK00370
TRK00380
TRK00390
TRK00400
TRK00410
TRK00420
TRK00430

```

```

SUBROUTINE TRNCLD(XNORM, TIME, ACLWAK, ACLSPH)
ROUTINE FOR DETERMINING CONTRIBUTION FROM SPHERE AND WAKE
BEFORE THE BUOYANT SPHERE HAS CONVERTED TO THE WIND MODEL.
INPUTS
  XNORM  -DISTANCE BETWEEN THE TRANSMITTER AND RECEIVER
  TIME   -TIME AT WHICH TRANSMITTANCE IS DESIRED
OUTPUTS
  ACLWAK -CONTRIBUTION FROM WAKE TO OPTICALLY WEIGHTED CONCENTRATION
          ALONG GIVEN LINE OF SIGHT
  ACLSPH -CONTRIBUTION FROM BUOYANT SPHERE TO OPTICALLY WEIGHTED
          CONCENTRATION ALONG GIVEN LINE OF SIGHT
SUBROUTINES NEEDED
  AVRQ   -COMPUTE THE AVERAGE OF THE MOMENTS FOR THE DISCS
  WIN    -COMPUTE THE WIND SPEED AT A GIVEN HEIGHT
  AMOUNT -COMPUTE THE DISTRIBUTION OF THE LOADING BETWEEN THE BUOYANT
          SPHERE AND WAKE.
  CONLEN -COMPUTE THE LENGTH OF INTERSECTION OF NON-HORIZONTAL LINE
          OF SIGHT WITH A CONICAL SHAPED WAKE
  TRNCHK -ROUTINE TO CHECK IF THE OBSCURATION IS SUCH THAT THE
          TRANSMITTANCE IS LESS THAN A USET SPECIFIED AMOUNT
*****
LOGICAL HORIZ, TEST
DIMENSION CENTER(3)
COMMON/MODE/HORIZ
COMMON/GEOM/COSTH2, SINTH, SINTH2, VISEXT, RTPI, SCRN(2)
COMMON/BUOYCL/RSPH, DELT, VZ, XCM, YCM, ZCM, XTOP, YTOP, SPHNS(3), RISTIM
COMMON/PRTINF/R0, VGRAV(3), NPRTS
COMMON/DISCS/NDSCS, TDSC(20), XDSC(20), ZDSC(20), R2DSC(20), QDSC(20,3)
COMMON/MOS/DIFF(2,200), NCHTOT, PRSEP(200), NTOT, NARY, ITOT,
+ DMM(600), DMMY(401)
COMMON /IUNIT/IOIN, IOOUT, IPHFUN, LOUNIT, NDIRTU, NCLINT, KSTOR, NPLOTUT
COMMON/LOS/TR(3), RE(3), U(3)
COMMON/TRANNY/THRESH, TEST, NWL, NSOIL
COMMON/ACL/CWINDS, CWINDC, CWINDW
COMMON /CONST/PI, PI2, PIRAD, TWOPI, TORRMB, CDEGK
ACLWAK=0.0
ACLSPH=0.0
DETERMINE THE RADIUS OF THE BASE OF THE CONE
  ZX=5.0
  CALL AVRQ(ZX, TIME, QTOT, XBAVG, SIG2X, SIG2Y)
  IF(QTOT.LT.1.E-10)GO TO 33
  SIGX=SQRT(SIG2X)
  SIGY=SQRT(SIG2Y)
  RB=SQRT(SIGX*SIGY)
  GO TO 35
33 CALL WIN(5.0, UW, V)
  XBAVG=UW*TIME
  RB=0.0
35 VOLWAK=(PI/3.)*(ZCM-5.0)*(RSPH**2+RSPH*RB+RB**2)
  VOLSPH=(4./3.)*PI*RSPH**3
  CALL AMOUNT(VOLSPH, WAKAL, SPHAL)
  DO 80 J=1, ITOT
  IF (VOLWAK.LE.0.0) GO TO 68
  XB=XBAVG+DIFF(1,J)

```

	YB=DIFF(2,J)	TRD00730
	XC=XCM+DIFF(1,J)	TRD00740
	YC=YCM+DIFF(2,J)	TRD00750
	IF(ABS(U(3)).LE..01)GO TO 110	TRD00760
	ZBOT=5.0	TRD00770
	CALL CONLEN(U,TR,ZCM,ZBOT,XC,YC,RSPH,RB,XB,YB,XNORM,PLENWK)	TRD00780
	ACLW=PLENWK*WAKAL/VOLWAK	TRD00790
	GO TO 69	TRD00800
68	ACLW=0.0	TRD00810
69	CONTINUE	TRD00820
	ACLWAK=ACLWAK+ACLW	TRD00830
	CALL TRNCHK(CWINDS,ACLWAK,ACLSPH)	TRD00840
	IF(TEST)GO TO 999	TRD00850
C		TRD00860
C	DETERMINE CONTRIBUTION FROM SPHERE FO A SLANT PATH	TRD00870
C		TRD00880
	CENTER(1)=XCM+DIFF(1,J)-TR(1)	TRD00890
	CENTER(2)=YCM+DIFF(2,J)-TR(2)	TRD00900
	CENTER(3)=ZCM-TR(3)	TRD00910
	CLOSE=U(1)*CENTER(1)+U(2)*CENTER(2)+U(3)*CENTER(3)	TRD00920
	CON=CENTER(1)**2+CENTER(2)**2+CENTER(3)**2-RSPH**2	TRD00930
	RADIC=CLOSE**2-CON	TRD00940
	IF(RADIC.LT.0.0)GO TO 75	TRD00950
	PNEAR=CLOSE-SQRT(RADIC)	TRD00960
	PFAR=CLOSE+SQRT(RADIC)	TRD00970
	PLENSP=AMIN1(PFAR,XNORM)-AMAX1(PNEAR,0.0)	TRD00980
	IF(PLENSP.LT.0.0)PLENSP=0.0	TRD00990
	ACLS=PLENSP*SPHAL/VOLSPH	TRD01000
	GO TO 76	TRD01010
75	ACLS=0.0	TRD01020
76	ACLSPH=ACLSPH+ACLS	TRD01030
	CALL TRNCHK(CWINDS,ACLWAK,ACLSPH)	TRD01040
	IF(TEST)GO TO 999	TRD01050
	GO TO 80	TRD01060
C		TRD01070
C	COMPUTE CONTRIBUTIONS FOR SPHERE AND WAKE FOR A HORIZONTAL PATH	TRD01080
C		TRD01090
110	IF(TR(3).GT.ZCM+RSPH)GO TO 999	TRD01100
	XCEN=XCM+DIFF(1,J)	TRD01110
	YCEN=YCM+DIFF(2,J)	TRD01120
	IF(TR(3).LT.ZCM-RSPH)GO TO 130	TRD01130
	RADIUS=SQRT(RSPH**2-(TR(3)-ZCM)**2)	TRD01140
	CALL PATH(TR,U,XCEN,YCEN,RADIUS,PLENSP)	TRD01150
	IF(PLENSP.LT.0.0)PLENSP=0.0	TRD01160
	ACLSPH=ACLSPH+PLENSP*SPHAL/VOLSPH	TRD01170
	CALL TRNCHK(CWINDS,ACLWAK,ACLSPH)	TRD01180
	IF(TEST)GO TO 999	TRD01190
130	IF(TR(3).GT.ZCM)GO TO 999	TRD01200
	IF(TR(3).LE.5.0)GO TO 999	TRD01210
	ZETA=(TR(3)-5.0)/(ZCM-5.0)	TRD01220
	XCEN=ZETA*XCEN+(1-ZETA)*XB	TRD01230
	YCEN=ZETA*YCEN+(1-ZETA)*YB	TRD01240
	RADIUS=ZETA*RSPH+(1-ZETA)*RB	TRD01250
	CALL PATH(TR,U,XCEN,YCEN,RADIUS,PLENWK)	TRD01260
	IF(PLENWK.LT.0.0)PLENWK=0.0	TRD01270
	ACLWAK=ACLWAK+PLENWK*WAKAL/VOLWAK	TRD01280
	CALL TRNCHK(CWINDS,ACLWAK,ACLSPH)	TRD01290
	IF(TEST)GO TO 999	TRD01300
80	CONTINUE	TRD01310
999	RETURN	TRD01320
	END	TRD01330


```

SUBROUTINE UNIT(A,B,XNORM)
DIMENSION A(2),B(2)
C *** B IS THE NORM OF A, AND XNORM IS THE MAGNITUDE
XNORM=SQRT(A(1)**2+A(2)**2)
B(1)=A(1)/XNORM
B(2)=A(2)/XNORM
RETURN
END

```

```

UNIT0010
UNIT0020
UNIT0030
UNIT0040
UNIT0050
UNIT0060
UNIT0070
UNIT0080

```

```

SUBROUTINE VEHCL(NATMOS,ZTMP,TMPMES,ZWND,WNDMES,THWND,PHI,NSOIL,
1          SILT,NWL,TRNCOR,RECCOR,TIME,DHDT,V0,VEHDIR,
2          VEHSPD,VEHWID,VEHWHT,VEHTYP,NEWATM,NEWVEH,
3          TRNLOS,NERR)
C
C THIS ROUTINE CONTROLS THE FLOW OF THE CALCULATION FOR THE
C VEHICLE GENERATED DUST CLOUD.
C
C INPUTS
C SEE DRTAN FOR DETAILS
C
C OUTPUTS
C
C TRNLOS - TRANSMITTANCE ALONG THE LINE OF SIGHT
C
C NERR - ERROR CODE
C
C *****
C DIMENSION ZTMP(2),TMPMES(2),ZWND(2),WNDMES(2)
C DIMENSION TRNCOR(3),RECCOR(3),TRN(3),REC(3),TRNFRM(2,2),ORIG(2),
1 V1(2),V0(2),VDIR(2)
C LOGICAL DHDT,ERR,NEWATM,NEWVEH
C INTEGER VEHTYP
C COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLINT,KSTOR,NPLOTU
C COMMON/GEOM/COSTH2,SINTH,SINTH2,VISEXT,RTPI,SCRN(2)
C COMMON/WNDPRM/DXZ0,DYX0,DZ0,U0,UM,DN,ZINV
C COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK
C DATA RTPI /1.772453/
C THETAX=THWND*PIRAD
C IF(.NOT.NEWATM)GO TO 100
C CALL ATMCAL(NATMOS,ZTMP,TMPMES,ZWND,WNDMES,PHI,THETAX,DHDT,ERR)
C IF(.NOT.ERR)GO TO 100
C NERR=7
C GO TO 999
100 CONTINUE
C IF(.NOT.NEWVEH)GO TO 5
C CALL VSRCL(VEHSPD,VEHWID,VEHWHT,VEHTYP,NSOIL,SILT)
C CALL PREVEH(NSOIL,NWL)
5 CONTINUE
C
C COMPUTE DIRECTION VECTOR FOR THE VEHICLE MOTION FROM USERS INPUT
C
C ANGL=VEHDIR*PIRAD
C V1(1)=VEHSPD*COS(ANGL)
C V1(2)=VEHSPD*SIN(ANGL)
C
C COMPUTE THE ROTATION TRANSFORMATION MATRIX TO CONVERT THE USER
C DEFINED COORDINATES INTO LOCAL COORDINATES WITH THE X-AXIS IN
C THE WIND DIRECTION.
C
C TRNFRM(1,1)=COS(THETAX)
C TRNFRM(2,2)=TRNFRM(1,1)
C TRNFRM(1,2)=SIN(THETAX)
C TRNFRM(2,1)=-TRNFRM(1,2)
C ORIG(1)=V0(1)
C ORIG(2)=V0(2)
C
C COMPUTE NEW COORDINATES BY MULTIPLYING BY THE TRANSFORMATION MATRIX
C
C TRN(3)=TRNCOR(3)
C REC(3)=RECCOR(3)
C DO 20 I=1,2
C TRN(I)=0.0
C REC(I)=0.0
C VDIR(I)=0.0
C DO 10 J=1,2
C TRN(I)=TRN(I)+TRNFRM(I,J)*(TRNCOR(J)-ORIG(J))
C REC(I)=REC(I)+TRNFRM(I,J)*(RECCOR(J)-ORIG(J))
C VDIR(I)=VDIR(I)+TRNFRM(I,J)*V1(J)
C
C *****

```

```

10 CONTINUE
20 CONTINUE
C CALL VEHTRN ROUTINE TO USE COMPUTED QUADRATIC FITS TO CALCULATE
C A TRANSMITTANCE
C CALL VEHTRN(TRN,REC,TIME,VDIR,TRNLOS)
999 RETURN
END

```

```

VCL00710
VCL00720
VCL00730
VCL00740
VCL00750
VCL00760
VCL00770
VCL00780
VCL00790

```

```

SUBROUTINE VEHTRN( TRN, REC, TIME, VDIR, TRNLOS )
THIS ROUTINE PARAMETERIZES THE LINE CONNECTING THE TRANSMITTER
AND RECEIVER IN THE LOCAL COORDINATE SYSTEM AND DOES A TRAPEZOIDAL
INTEGRATION FROM VEHICLE TIME=0.0 TO VEHICLE TIME=TIME THE
TRANSMITTANCE IS DESIRED.
INPUTS
  TRN   - THE COORDINATES OF THE TRANSMITTER IN THE LOCAL COORDINATE
          SYSTEM
  REC   - THE COORDINATES OF THE RECEIVER IN THE LOCAL COORDINATE
          SYSTEM
  TIME  - THE PRESENT TIME AT WHICH A TRANSMITTANCE IS DESIRED
  NWL   - INTEGER INDEX FOR WAVELENGTH BEING USED
  NSOIL - INTEGER INDEX FOR SOIL TYPE
  VDIR  - VECTOR INDICATING THE DIRECTION AND SPEED OF THE VEHICLE
          IN THE LOCAL COORDINATE SYSTEM
OUTPUT
  TRNLOS - TRANSMITTANCE ALONG THE LINE OF SIGHT AT THE INDICATED
           TIME
FUNCTIONS AND SUBROUTINES NEEDED
  GRAND EVALUATES THE INTEGRAND
*****
LOGICAL TEST
DIMENSION TRN(3), REC(3), TR(3), RE(3), VDIR(2), DWF(5,2), U(3)
COMMON/ M05/ DMMY(604), DMM(600),
+          ICOUNT, TIMES(25), XC0(3,25), XC1(3,25), RT(3,25),
+          RB(3,25), Z2(3,25)
COMMON/ TRANNY/ THRESH, TEST, NWL, NSOIL
DATA DWF/ 1., .93, .52, .44, 2.E-03, 1., 1., 1., 1., 4.E-03/
TEST=.FALSE.
PARAMETERIZE THE LINE CONNECTING THE TRANSMITTER AND RECEIVER
  XNORM=0.0
  DO 10 I=1,3
    RE(I)=REC(I)
    TR(I)=TRN(I)
    U(I)=RE(I)-TR(I)
    XNORM=XNORM+U(I)**2
10 CONTINUE
  XNORM=SQRT(XNORM)
  U(1)=U(1)/XNORM
  U(2)=U(2)/XNORM
  U(3)=U(3)/XNORM
  INCREMENT VEHICLE TRAVEL TIME AND CALL GRAND TO COMPUTE THE VALUE
  OF THE INTEGRAND. IF THE VALUE OF THE VARIABLE TIME IS GREATER
  THAN THE MAXIMUM TIME THAT HAS BEEN STORED (APPROX. 373 SEC) THEN
  ANY DUST PRODUCED MORE THAN TIMES(20) SECONDS IS ASSUMED TO HAVE
  NO EFFECT ON THE TRANSMITTANCE.
  IF( TIME.LE. TIMES( ICOUNT ) ) GO TO 11
  TINC=TIMES( ICOUNT )/400.
  TSTART=TIME-TIMES( ICOUNT )
  GO TO 15
11 TINC=TIME/400.
  TSTART=0.0

```

```

VTN00340
VTN00010
VTN00020
VTN00030
VTN00040
VTN00050
VTN00060
VTN00070
VTN00080
VTN00090
VTN00100
VTN00110
VTN00120
VTN00130
VTN00140
VTN00150
VTN00160
VTN00170
VTN00180
VTN00190
VTN00200
VTN00210
VTN00220
VTN00230
VTN00240
VTN00250
VTN00260
VTN00270
VTN00280
VTN00290
VTN00300
VTN00310
VTN00320
VTN00330
VTN00350
VTN00360
VTN00370
VTN00380
VTN00390
VTN00400
VTN00410
VTN00420
VTN00430
VTN00440
VTN00450
VTN00460
VTN00470
VTN00480
VTN00490
VTN00500
VTN00510
VTN00520
VTN00530
VTN00540
VTN00550
VTN00560
VTN00570
VTN00580
VTN00590
VTN00600
VTN00610
VTN00620
VTN00630
VTN00640
VTN00650
VTN00660
VTN00670
VTN00680
VTN00690
VTN00700

```

```

15 SUM=0.0
   DO 50 I=1,401
   TIVEH=TSTART+FLOAT(I-1)*TINC
   CALL GRAND(U,TR,XNORM,TIME,TIVEH,VDIR,VALUE)
   IF(I.EQ.1.OR.I.EQ.401)GO TO 20
   SUM=SUM+VALUE
   GO TO 40
20 SUM=SUM+VALUE/2.
40 CONTINUE
   SUM1=SUM*TINC
   ACLWAK=0.0
   ACLSPH=0.0
   CALL TRNCHK(SUM1,ACLWAK,ACLSPH)
   IF(TEST)GO TO 998
50 CONTINUE
   SUM=SUM*TINC
   ACL=OWF(NWL,NSOIL)*SUM
   TRNLOS=EXP(-ACL)
   GO TO 999
998 TRNLOS=0.0
999 RETURN
   END

```

```

VTN00710
VTN00720
VTN00730
VTN00740
VTN00750
VTN00760
VTN00770
VTN00780
VTN00790
VTN00800
VTN00810
VTN00820
VTN00830
VTN00840
VTN00850
VTN00860
VTN00870
VTN00880
VTN00890
VTN00900
VTN00910
VTN00920

```

```

SUBROUTINE VSRCK(VSPD,VWID,VWHT,VEHTYP,NSOIL,SILT)
THIS SUBROUTINE INITIALIZES THE DUST CLOUD PRODUCED BY A VEHICLE
INPUTS
  VSPD - THE VELOCITY OF THE VEHICLE
  VWID - WIDTH OF THE VEHICLE
  VWHT - WEIGHT OF THE VEHICLE IN KGS.
  NSOIL - SOIL TYPE
  SILT - SILT CONTENT OF THE SOIL
OUTPUTS
  STOPED IN COMMON/DISCS/ AND COMMON/PRE/
  NDSCS - NUMBER OF DISCS (FOR A VEHICLE ONLY ONE)
  TDSC - TIME OF RELEASE OF THE DISCS
  XDSC - X POSITION OF DISC AT TIME OF RELEASE
  ZDSC - HEIGHT OF RELEASE OF THE DISC
  R2DSC - SQUARE OF THE RADIUS OF THE DISC
*****
  INTEGER VEHTYP
  COMMON/PRTINF/R0,VGRAY(3),NPRTS
  COMMON /IDUNIT/IDIN,IDOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
  COMMON/DISCS/NDSCS,TDSC(20),XDSC(20),ZDSC(20),R2DSC(20),QDSC(20,3)
  COMMON/WNDPRM/DXZ0,DYX0,DZ0,U0,UM,DN,ZINV
  COMMON/EKTEMP/Z0,ZL,T0,TC1,TC2,TC3
  COMMON/PRE/Z,RT2DZ
  COMMON/VL/VLOAD
  NDSCS=1
  NPRTS=1
  VGRAY(1)=0.0
  QDSC(1,1)=1.0
  INITIALIZE THE VEHICLE SOURCE FOR WHEELED VEHICLES
    ZDSC(1)=VWID/8.
    Z=VWID/4.
    ZZ=VWID/4.
    DZ=DIFFUS(Z0,ZL,ZZ)
    DX=DXZ0*DZ
    TDSC(1)=-9.*(VWID**2)/512./DZ
    TOF=-TDSC(1)
    CALL MOMENT(VGRAY,Z,ZDSC(1),TOF,Q,XBAR,SIGW2,SIGP2)
    XDSC(1)=-XBAR
    A=1.0
    B=SIGW2+SIGP2
    C=4.*(C(SIGW2*SIGP2)-((VWID/3.))**4))
    RAD=B**2-A*C
    R2DSC(1)=-B+SQRT(RAD)
    RT2DZ=SQRT(2.*DZ)
  INITIALIZE LOADING FOR VEHICLE (VLOAD IN KG/SEC)
20 SILTPC=100.*SILT
  A=3.8E-9
  IF(VEHTYP.GT.0)A=1.52E-08
  Q=A*VSPD*VWHT*SILTPC
  ALPHA=240.

```

999 VLOAD=ALPHA*VSPD*Q
RETURN
END

VRS00710
VRS00720
VRS00730

```

SUBROUTINE VSUM(A,B,S,C)
DIMENSION A(2),B(2),C(2)
C *** C=A+S*B WHERE A,B,C ARE VECTORS AND S IS SCALAR
DO 14 J=1,2
14 C(J)=A(J)+S*B(J)
RETURN
END

```

```

VSU00010
VSU00020
VSU00030
VSU00040
VSU00050
VSU00060
VSU00070

```


<pre> SUBROUTINE WIN(Z,U,V) COMMON/STARS/USTAR,TSTAR,ZSTAR COMMON/EKWIND/ALP,C,PYF,PF, UHAT,VHAT COMMON/EKTEMP/Z0,ZL,T0,TC1,TC2,TC3 COMMON/BUOYCL/Y(8),SPHNS(3),RISTIM ***** PURPOSE TO COMPUTE THE WIND SPEED AT A SPECIFIED HEIGHT INPUTS Z HEIGHT AT WHICH WIND SPEEDS ARE DESIRED OUTPUTS U WIND SPEED IN THE DIRECTION OF THE GROUND WIND V WIND SPEED PERPENDICULAR TO THE GROUND WIND CALLED BY DIFEQ SUBROUTINES AND FUNCTIONS NEEDED WNDCL CALCULATES SCALED WIND SPEED ***** IF(Z.GT.ZSTAR)GO TO 100 U=USTAR*WNDCL(Z0,ZL,Z) V=0.0 GO TO 999 100 UE=C*EXP(-ALP*Z)*COS(ALP*Z)-PYF VE=-C*EXP(-ALP*Z)*SIN(ALP*Z)+PF U=UHAT*UE+UHAT*VE V=-VHAT*UE+UHAT*VE 999 RETURN END </pre>	<pre> WIN00100 WIN00200 WIN00300 WIN00400 WIN00500 WIN00600 WIN00700 WIN00800 WIN00900 WIN01000 WIN01100 WIN01200 WIN01300 WIN01400 WIN01500 WIN01600 WIN01700 WIN01800 WIN01900 WIN02000 WIN02100 WIN02200 WIN02300 WIN02400 WIN02500 WIN02600 WIN02700 WIN02800 WIN02900 WIN03000 WIN03100 WIN03200 WIN03300 WIN03400 WIN03500 WIN03600 WIN03700 WIN03800 </pre>
--	--

```

C      FUNCTION WNDCAL(Z0,ZL,Z)
C      *****
C      PURPOSE
C      TO CALCULATE THE WIND SPEED, U/U*, SCALED BY THE FRICTION
C      VELOCITY FROM GIVEN FRICTION HEIGHT AND MONIN-OBUKHOV LENGTH AT A
C      SPECIFIED HEIGHT.
C
C      INPUTS
C      Z0      THE FRICTION HEIGHT IN METERS.
C      ZL      THE MONIN-OBUKHOV LENGTH IN METERS.
C      Z       THE HEIGHT AT WHICH THE SCALED VELOCITY IS DESIRED
C              IN METERS
C
C      RETURNS VELOCITY SCALED BY FRICTION VELOCITY
C
C      CALLED BY ATMICAL,WIN AND RISE
C      *****
C      LOGICAL LOW
C      COMMON /COEF/AM,CW,BW,DW,AT,CT,BT,DT
C      COMMON /IDUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
C      PSIM(Z,Z1)=ALOG((1.-Z)/(1.-Z1))-ALOG((1.+Z)/(1.+Z1))+
C      $2.*(ATAN(Z)-ATAN(Z1))
C      PSIMS(Z)=-7.*Z
C      PHIM(Z)=(1.-16.*Z)**(-.25)
C
C      PHIM      THE SHEAR OF MOMENTUM
C      PSIM      THE UNIVERSAL FUNCTION FOR THE DEVIATION FROM
C      LOGARITHMIC WIND VELOCITY BOUNDARY LAYER PROFILE IN AN
C      UNSTABLE ATMOSPHERE
C      PSIMS     THE SAME AS PSIM FOR A STABLE ATMOSPHERE
C
C      IF(ABS(ZL).LE.1.E3)GO TO 100
C      WNDCAL=ALOG(1.+Z/Z0)
C      GO TO 999
100  CONTINUE
C      P=SIGN(1.,ZL)
C      LOW=.TRUE.
C      S=Z/ZL
C      IF(S.LE.1.5.AND.S.GE.-2.)GO TO 10
C      S=AMIN1(S,1.5)
C      S=AMAX1(S,-2.)
C      LOW=.FALSE.
10  CONTINUE
C      IF(P)120,130,130
120  S=1./PHIM(S)
C      S1=Z0/ZL
C      S0=1./PHIM(S1)
C      WNDCAL=PSIM(S,S0)
C
C      DETERMINE THE CONSTANTS FOR MATCHING AT Z/ZL=-2.
C
C      S2=-2.
C      AW=-3.*(1.-16.*(S2)**(-.25))*((-S2)**(1./3.))
C      CW=-1.*AW*(-S2)**(-1./3.)
C      GO TO 52
130  CONTINUE
C      PSI=PSIMS(S)
C      WNDCAL=ALOG(1.+S*ZL/Z0)-PSI
C
C      FIND THE CONSTANTS FOR MATCHING OF STABLE PROFILE AT Z/ZL=1.5

```

```

WND00010
WND00020
WND00030
WND00040
WND00050
WND00060
WND00070
WND00080
WND00090
WND00100
WND00110
WND00120
WND00130
WND00140
WND00150
WND00160
WND00170
WND00180
WND00190
WND00200
WND00210
WND00220
WND00230
WND00240
WND00250
WND00260
WND00270
WND00280
WND00290
WND00300
WND00310
WND00320
WND00330
WND00340
WND00350
WND00360
WND00370
WND00380
WND00390
WND00400
WND00410
WND00420
WND00430
WND00440
WND00450
WND00460
WND00470
WND00480
WND00490
WND00500
WND00510
WND00520
WND00530
WND00540
WND00550
WND00560
WND00570
WND00580
WND00590
WND00600
WND00610
WND00620
WND00630
WND00640
WND00650
WND00660
WND00670
WND00680
WND00690
WND00700

```

```

S2=1.5
BW=1./(<Z0/ZL+S2>)+7.
DW=-1.*S2*BW
52 CONTINUE
IF(<LOW>)GO TO 999
IF(<P>)53,53,54
53 WNDAL=WNDAL+CW+AW*(-ZL/Z)**(<1./3.>)
GO TO 999
54 WNDAL=WNDAL+DW+BW*Z/ZL
999 WNDAL=WNDAL/.4
998 CONTINUE
RETURN
END

```

```

WND00710
WND00720
WND00730
WND00740
WND00750
WND00760
WND00770
WND00780
WND00790
WND00800
WND00810
WND00820
WND00830

```

```

SUBROUTINE NMMW(FREQGH,ICLMAT,MMTRAN,IERR)
COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK
COMMON /CLYMAT/TEMP,PRESS,RH,AH,DP,VIS,CLDAMT,CLDHYT,
+ FOGPRB,WNDVEL,WNDDIR,IPASCT
COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLINT,KSTOR,NPLOTUN
COMMON /GEOMET/PTS<15>,IGEOSW
PURPOSE: TO CALCULATE THE EXTINCTION AND BACKSCATTER AT MILLIMETER
FREQUENCIES 10 TO 1000 GHZ, DUE TO
GASEOUS ABSORPTION, FOG<ICE AND WATER> AND
CLOUD BULK ATTENUATION, RAIN, SNOW EXTINCTION.

** INPUT TO THE NMMW MODULE IS PERFORMED THROUGH A CARD ORDER-
** INDEPENDENT INPUT TECHNIQUE. A FOUR-LETTER IDENTIFIER IN COLS.
** 1-4 OF EACH RECORD SPECIFIES THE TYPE OF DATA BEING READ BY THE
** MODULE. THE INPUT CARDS MAY APPEAR IN ANY ORDER WITH THE EXCEPTION
** OF OF THE <GO> END OF READ SENTINEL, WHICH MUST BE THE LAST CARD
** READ. ALL OF THE FOLLOWING CARDS ARE READ IN UNDER THE FORMAT
** <A4,1X,3(E10.4,1X)> :

CARD IDENTIFIER : PATH
VARIABLES READ : MMWPTH <PATH LENGTH <KM>>

CARD IDENTIFIER : ATMO
VARIABLES READ : TEMP1 <TEMPERATURE <DEG C>>
PRESS1 <PRESSURE <MB>>
ABSHUM <IF .GT. 0, ABSOLUTE
HUMIDITY <GM/M**3>>
<IF .LT. 0, RELATIVE
HUMIDITY <%>>

***NOTE: TEMP1,PRESS1,ABSHUM WILL BE PASSED FROM CLIMAT IF
ICLMAT=1. IN THAT EVENT, THE <ATMO> CARD IS NOT NEEDED.

CARD IDENTIFIER : FOGD
VARIABLES READ : FOGDEN <FOG DENSITY <LIQUID WATER,
GM/M**3>>

CARD IDENTIFIER : RAIN
VARIABLES READ : RAINRT <RAIN RATE <MM/HR>>

CARD IDENTIFIER : SNOW
VARIABLES READ : SNOWRT <SNOW RATE <MM/HR> WATER EQUIV.>

CARD IDENTIFIER : GO
VARIABLES READ : NONE <END OF READ SENTINEL>

MAIN ROUTINE CALLS NMMW. NMMW THEN CALLS
(1)MMWGS<GAS ABSORPTION>, (2)MMWFG<FOG ABSORPTION,BACKSCATTER>,
(3)MMWRN<RAIN EXTINCTION,BACKSCATTER>,
(4)MMSNO<SNOW EXTINCTION,BACKSCATTER>, AND RETURNS TO MAIN NMMW

LOCAL VARIABLES
REAL MMWPTH,MMTRAN,MMBSXS
DIMENSION DAT<3>,IAL<6>
DATA IAL/2HPA ,2HAT ,2HFO ,2HRA ,2HSN ,2HGO /
IERR=0
GASABS=0.
FOGEXT=0.
RAINEX=0.
FOGBS=0.
RAINBS=0.
SNOWBS=0.
5 READ<IOIN,400>IALFA,IALFA2,<DAT<L>,L=1,3>
400 FORMAT<2A2,1X,3(E10.4,1X)>
IF<IALFA.EQ.IAL<1>> GO TO 10
IF<IALFA.EQ.IAL<2>> GO TO 20

```

IF(IALFA.EQ.IAL(3)) GO TO 30	MMM00710
IF(IALFA.EQ.IAL(4)) GO TO 40	MMM00720
IF(IALFA.EQ.IAL(5)) GO TO 50	MMM00730
IF(IALFA.EQ.IAL(6)) GO TO 60	MMM00740
WRITE(IOOUT,450)IALFA,IALFA2	MMM00750
450 FORMAT(1H0,20X,2A2,22H IS AN INCORRECT LABEL ///	MMM00760
GO TO 300	MMM00770
10 MMWPTH=DAT(1)	MMM00780
GO TO 5	MMM00790
20 TEMP1=DAT(1)	MMM00800
PRESS1=DAT(2)	MMM00810
ABSHUM=DAT(3)	MMM00820
GO TO 5	MMM00830
30 FOGDEN=DAT(1)	MMM00840
GO TO 5	MMM00850
40 RAINRT=DAT(1)	MMM00860
GO TO 5	MMM00870
50 SNOWRT=DAT(1)	MMM00880
GO TO 5	MMM00890
60 CONTINUE	MMM00900
IF(IGEOSW.NE.1)GO TO 99	MMM00910
MMWPTH=SQR((PTS(4)-PTS(1))**2+(PTS(5)-PTS(2))**2+	MMM00920
+(PTS(6)-PTS(3))**2)	MMM00930
99 CONTINUE	MMM00940
IF(ICLMAT.EQ.1) TEMP1=TEMP	MMM00950
IF(ICLMAT.EQ.1) PRESS1=PRESS	MMM00960
IF(ICLMAT.EQ.1) ABSHUM=AH	MMM00970
WRITE(IOOUT,500) TEMP1,PRESS1,ABSHUM,FOGDEN	MMM00980
WRITE(IOOUT,600) RAINRT,SNOWRT,FREQGH,MMWPTH	MMM00990
C CHECK MODEL INPUTS FOR RANGE OF VALIDITY.	MMM01000
IF(PRESS1.GE.500.) GO TO 100	MMM01010
IERR=1	MMM01020
WRITE(IOOUT,800)	MMM01030
100 IF(FREQGH.LE.1000..AND.FREQGH.GE.10.) GO TO 150	MMM01040
IERR=1	MMM01050
WRITE(IOOUT,900)	MMM01060
150 IF(IERR.EQ.1) GO TO 300	MMM01070
C CHANGE UNITS	MMM01080
PRSTOR=PRESS1/TORRMB	MMM01090
TEMPDK=TEMP1+CDEGK	MMM01100
C CALL INDIVIDUAL MODULES FOR GAS, FOG/CLOUD, SNOW, RAIN EXTINCTION	MMM01110
CALL MMWGS(TEMPDK,PRSTOR,ABSHUM,FREQGH,GASABS)	MMM01120
IF(FOGDEN.GT.1.E-10)	MMM01130
+ CALL MMWFG(FOGDEN,TEMPDK,FREQGH,FOGEXT,FOGBS)	MMM01140
+ IF(RAINRT.GT.1.E-10)	MMM01150
+ CALL MMRAN(RAINRT,TEMPDK,FREQGH,2.,RAINEX,RAINBS)	MMM01160
+ IF(SNOWRT.GT.1.E-10)	MMM01170
+ CALL MMSNO(SNOWRT,TEMPDK,FREQGH,SNOWEX,SNOWBS)	MMM01180
C COMPUTE TRANSMISSION	MMM01190
TOTEXT=GASABS+FOGEXT+RAINEX+SNOWEX	MMM01200
MMBSXS=FOGBS+RAINBS+SNOWBS	MMM01210
MMTRAN=EXP(-MMWPTH*TOTEXT)	MMM01220
C CHANGE UNITS FOR ABSORPTION/EXTINCTION FROM 1/KM TO DB/KM.	MMM01230
DBKM=4.343	MMM01240
GASABS=GASABS*DBKM	MMM01250
FOGEXT=FOGEXT*DBKM	MMM01260
RAINEX=RAINEX*DBKM	MMM01270
SNOWEX=SNOWEX*DBKM	MMM01280
WRITE(IOOUT,700) GASABS,FOGEXT,RAINEX,SNOWEX,MMTRAN	MMM01290
WRITE(IOOUT,750) FOGBS,RAINBS,SNOWBS,MMBSXS	MMM01300
C COMPUTATION COMPLETED	MMM01310
300 RETURN	MMM01320
C	MMM01330
500 FORMAT(1H0,///,47X,12HTEMPERATURE ,14X,F8.3,	MMM01340
+ 10H DEGREES C,/,47X,9HPRESSURE ,17X,F8.3,	MMM01350
+ 3H MB,/,47X,17HABSOLUTE HUMIDITY,9X,F8.3,	MMM01360
+ 7H G/M**3/,47X,11HFOG DENSITY,15X,F8.3,	MMM01370
+ 7H G/M**3)	MMM01380
600 FORMAT(1H ,46X,9HRAIN RATE,17X,F8.3,6H MM/HR,/,	MMM01390
+ ,47X,9HSNOW RATE,17X,F8.3,6H MM/HR,/,	MMM01400

	+	,47X,9HFREQUENCY,17X,F8.3,4H GHZ/	NMM01410
	+	,47X,11HPATH LENGTH,15X,F8.3,3H KM)	NMM01420
700	FORMAT	(1H0,46X,14HGAS ABSORPTION,10X,E10.4,6H DB/KM,/	NMM01430
	+	,47X,15HFOG EXTINCTION,9X,E10.4,6H DB/KM,/	NMM01440
	+	,47X,16HRAIN EXTINCTION,8X,E10.4,6H DB/KM,/	NMM01450
	+	,47X,16HSNOW EXTINCTION,8X,E10.4,6H DB/KM,/	NMM01460
	+	,47X,13HTRANSMISSION,11X,E10.4/)	NMM01470
750	FORMAT	(1H,46X,15HFOG BACKSCATTER,9X,E10.4,10H M**2/M**3,/	NMM01480
	+	,47X,16HRAIN BACKSCATTER,8X,E10.4,10H M**2/M**3,/	NMM01490
	+	,47X,16HSNOW BACKSCATTER,8X,E10.4,10H M**2/M**3,/	NMM01500
	+	,47X,17HTOTAL BACKSCATTER,7X,E10.4,10H M**2/M**3)	NMM01510
800	FORMAT	(1H0,47X,41HPRESSURE LESS THAN 500 MB, GAS ABSORPTION,	NMM01520
	+	19H WILL BE INACCURATE)	NMM01530
900	FORMAT	(1H0,47X,31HFREQUENCY<10. GHZ, OR >1000 GHZ,	NMM01540
	+	42H CALULATION WILL FAIL, USE OTHER FREQUENCY)	NMM01550
	END		NMM01560

```
FUNCTION AB(WA,A,CE,B,C)  
AB=A*EXP(-ABS((ALOG10(1.E4*WA/CE)/B))**C)  
RETURN  
END
```

```
AB 00010  
AB 00020  
AB 00030  
AB 00040
```

FUNCTION DOP(WA,A,CE1,B,C,CE2,D,E,CE3,F,G)	DOP00010
V=1./WA	DOP00020
V2=V*V	DOP00030
H1=CE1*CE1-V2	DOP00040
H2=CE2*CE2-V2	DOP00050
H3=CE3*CE3-V2	DOP00060
DOP=SQRT(A+B*H1/(H1*H1+C*V2)+D*H2/(H2*H2+E*V2)+F*H3/(H3*H3+G*V2))	DOP00070
RETURN	DOP00080
END	DOP00090

	SUBROUTINE INTRP(A,B,T,F,TT,FF,AA,BB,J)	INT00010
		INT00020
C	PURPOSE: TO DO FREQUENCY AND TEMPERATURE INTERPOLATION	INT00030
C	DIMENSION A(9,3), B(9,3), F(9), T(3)	INT00040
C	IF(TT,LT,T(1)) TT=T(1)	INT00050
	DO 11 J=2,3	INT00060
	IF(TT,LT,T(J)) GO TO 14	INT00070
11	CONTINUE	INT00080
	TT=T(3)	INT00090
	J=3	INT00100
14	CONTINUE	INT00110
	DO 15 I=2,9	INT00120
	IF(FF,LT,F(I)) GO TO 16	INT00130
15	CONTINUE	INT00140
	FF=F(9)	INT00150
	I=9	INT00160
16	FF=ALOG10(FF)	INT00170
	F0=ALOG10(F(I))	INT00180
	F1=ALOG10(F(I-1))	INT00190
	FF0=(F0-FF)/(F0-F1)	INT00200
	FF1=(FF-F1)/(F0-F1)	INT00210
	TF0=(T(J)-TT)/(T(J)-T(J-1))	INT00220
	TF1=(TT-T(J-1))/(T(J)-T(J-1))	INT00230
	A11=ALOG10(A(I-1,J-1))	INT00240
	A01=ALOG10(A(I,J-1))	INT00250
	A10=ALOG10(A(I-1,J))	INT00260
	A00=ALOG10(A(I,J))	INT00270
	APJ1=A11*FF0+A01*FF1	INT00280
	APJ0=A10*FF0+A00*FF1	INT00290
	AA=APJ1*TF0+APJ0*TF1	INT00300
	B11=ALOG10(B(I-1,J-1))	INT00310
	B10=ALOG10(B(I-1,J))	INT00320
	B01=ALOG10(B(I,J-1))	INT00330
	B00=ALOG10(B(I,J))	INT00340
	BPJ1=B11*FF0+B01*FF1	INT00350
	BPJ0=B10*FF0+B00*FF1	INT00360
	BB=BPJ1*TF0+BPJ0*TF1	INT00370
	AA=10.**AA	INT00380
	BB=10.**BB	INT00390
	RETURN	INT00400
	END	INT00410
		INT00420
		INT00430
		INT00440

	SUBROUTINE MMH20(V,T,PTOT,PH20,DATH20,ABH20)	MMH00010
C	*****	MMH00020
C	ROUTINE TO CALCULATE H2O VAPOR ABSORPTION FOR 0 TO 350 GHZ.	MMH00030
C	INPUTS ARE: WAVENUMBER(/CM), TEMPERATURE(KELVIN), TOTAL	MMH00040
C	PRESSURE(TORR), H2O VAPOR PRESSURE(TORR), LINE DATA ARRAY.	MMH00050
C	OUTPUT IS: H2O VAPOR ABSORPTION	MMH00060
C	CALL BY MMWGS MAKES NO CALLS	MMH00070
C	LOCAL VARIABLES:	MMH00080
C	WCD VAPOR COLUMN DENSITY(/CM/KM)	MMH00090
C	CT LINE STRENGTH TEMPERATURE CORRECTION	MMH00100
C	CA LINE WIDTH SELF BROADENING AND TEMP. CORRECTION	MMH00110
C	SA CORRECTED LINE STRENGTH	MMH00120
C	GA CORRECTED LINE WIDTH	MMH00130
C	ABS SINGLE LINE ABSORPTION(/KM)	MMH00140
C	*****	MMH00150
C	DIMENSION DATH20(37,4)	MMH00160
C	ABH20=0.	MMH00170
C	WCD=7.33994E26*PH20/760./T*(PFR(T))	MMH00180
C	CT=4.860773E-3*(T-296.)/T	MMH00190
C	CA=((296./T)**.62)*(PTOT+(4.*PH20))/760.	MMH00200
C	DO 500 L=1,37	MMH00210
	SA=DATH20(L,2)*WCD*EXP(DATH20(L,4)*CT)	MMH00220
	GA=DATH20(L,3)*CA	MMH00230
	ABS=SA*SUPK(V,DATH20(L,1),GA)	MMH00240
	ABH20=ABH20+ABS	MMH00250
500	CONTINUE	MMH00260
	RETURN	MMH00270
	END	MMH00280
		MMH00290
		MMH00300
		MMH00310
		MMH00320
		MMH00330
		MMH00340
		MMH00350

```

C      SUBROUTINE MMIDX(XL,T,ICE,H2OAB,H2OK2)
REF: RAY, APPLIED OPTICS, VOL. 11, P. 1836, (1972)
COMPLEX CINDX,XK
XXL=XL/10.
TT=T-273.16
T3=TT-25.
IF(ICE.NE.0) GO TO 150
C  PARAMETERS FOR WATER
100  EFIN=5.27137+.0216474*TT-1.31198E-3*TT*TT
ALFA=-16.8129/T+.0609265
XLS=3.3836E-4*EXP(2513.98/T)
SIGMA=1.25664E9
ES=78.54*(1.-4.579E-3*T3+1.19E-5*T3*T3-2.8E-8*T3*T3*T3)
GO TO 200
C
C  PARAMETERS FOR ICE
150  EFIN=3.168
ALFA=0.288+0.0052*TT+2.3E-4*TT*TT
XLS=9.990288E-5*EXP(1.32E4/(1.9869*T))
SIGMA=1.26*EXP(-1.25E4/(1.9869*T))
ES=203.168+2.5*TT+0.15*TT*TT
200  U=(ES-EFIN)*(XLS/XXL)**(1.-ALFA)
Y=1.+2.*(XLS/XXL)**(2.-2*ALFA)
1  EP=EFIN+((ES-EFIN)+U*SIN(ALFA*1.57079633))/Y
EPP=(U*COS(ALFA*1.57079633))/Y+SIGMA*XXL/1.88496E11
RE=SQRT((EP+SQRT(EP*EP+EPP*EPP))/2.)
AI=-EPP/2./RE
IF(ICE.NE.0) GO TO 400
C
IF(XXL.LE..034) GO TO 307
IF(XXL.GT..1) GO TO 311
306  R2=DOP(XXL,1.83899,1639.,52340.4,10399.2,588.24,345005.,
+ 259913.,161.29,43319.7,27661.2)
R2=R2+R2*T3*1.E-3*EXP((2.5E-5*XXL)**.25)
RE=RE*(XXL-.034)/.066+R2*(.1-XXL)/.066
GO TO 311
307  RE=DOP(XXL,1.83899,1639.,52340.4,10399.2,588.24,345005.,
+ 259913.,161.29,43319.7,27661.2)
RE=RE+RE*T3*1.E-3*EXP((2.5E-5*XXL)**.25)
311  CONTINUE
IF(XXL.GT..3) GO TO 500
AI=AI+AB(XXL,.25,300.,.47,3.)+AB(XXL,.39,17.,.45,1.3)
+ AB(XXL,.41,62.,.35,1.7)
GO TO 500
C
400  CONTINUE
IF(XXL.GT.0.08) GO TO 500
405  RC=DOP(XXL,1.225,1652.9,1.12082E6,46E-11,909.09,416441.,118852.,
+ 223.2,47031.8,126834.)
RE=RE*(XXL-0.02)/0.06+R2*(0.08-XXL)/0.06
AI=AI+AB(XXL,.242,62.,.23,1.6)+AB(XXL,.581,44.8,0.055,1.)
C
500  CINDX=CMPLX(RE,AI)
XK=(CINDX*CINDX-1)/(CINDX*CINDX+2)
H2OAB=AIMAG(-XK)
H2OK2=XK*CONJG(XK)
RETURN
END

```

```

MMI00010
MMI00020
MMI00030
MMI00040
MMI00050
MMI00060
MMI00070
MMI00080
MMI00090
MMI00100
MMI00110
MMI00120
MMI00130
MMI00140
MMI00150
MMI00160
MMI00170
MMI00180
MMI00190
MMI00200
MMI00210
MMI00220
MMI00230
MMI00240
MMI00250
MMI00260
MMI00270
MMI00280
MMI00290
MMI00300
MMI00310
MMI00320
MMI00330
MMI00340
MMI00350
MMI00360
MMI00370
MMI00380
MMI00390
MMI00400
MMI00410
MMI00420
MMI00430
MMI00440
MMI00450
MMI00460
MMI00470
MMI00480
MMI00490
MMI00500
MMI00510
MMI00520
MMI00530
MMI00540
MMI00550
MMI00560
MMI00570
MMI00580
MMI00590

```

```

SUBROUTINE MMXY(V,T,PTOT,PH2O,DATA02,ABS02)
*****
ROUTINE TO CALCULATE ABSORPTION DUE TO OXYGEN. METHOD IS THAT OF
LIEBE, GIMMESTAD, & HOPPONEN. IEEE TRANS. ANT. PROP. V.25, P327.
INPUTS ARE: FREQUENCY(GHZ), TEMPERATURE(KELVIN), TOTAL
PRESSURE(TORR), H2O VAPOR PRESSURE(TORR), O2 LINE DATA
ARRAY.
OUTPUTS ARE: O2 ABSORPTION (1/KM)
CALLED FROM MMWGS CALLS NO OTHER ROUTINES.
LOCAL VARIABLES:
  T2      300.2T
  PHI     TEMPERATURE CORRECTION FOR LINE STRENGTHS
  S        CORRECTED LINE STRENGTH(HZ TORR)
  GAMMA    CORRECTED LINE WIDTH (1/GHZ)
  XIF      LINE INTERFERENCE FACTOR
  VMI      DATA02(L,1)-V
  VPL      DATA02(L,1)+V
  PROFIL   MODIFIED VANVLECK-WEISSKOPF LINE SHAPE
-----
  DIMENSION DATA02(42,6)
  ABS02=0.
  T2=300.2T
  DO 500 L=1,42
    K=IFIX(DATA02(L,6))
    PHI=T2+T2+T2*EXP(-6.895E-3*K*(K+1)*(T2-1.))
    S=0.2095*PTOT*DATA02(L,2)*PHI
    GAMMA=DATA02(L,3)*(0.929*PTOT+T2**0.9+1.3*T2*PH2O)*1.E-3
    XIF=DATA02(L,4)*T2**DATA02(L,5)*PTOT*1.E-3
    VMI=DATA02(L,1)-V
    VPL=VMI+2*V
    PROFIL=(V/DATA02(L,1))*((GAMMA-VMI*XIF)/(VMI*VMI+GAMMA*GAMMA)+
      * (GAMMA-VPL*XIF)/(VPL*VPL+GAMMA*GAMMA))
    ABS02=ABS02+S*PROFIL*V*4.192E-5
  500 CONTINUE
  RETURN
  END

```

```

MM000010
MM000020
MM000030
MM000040
MM000050
MM000060
MM000070
MM000080
MM000090
MM000100
MM000110
MM000120
MM000130
MM000140
MM000150
MM000160
MM000170
MM000180
MM000190
MM000200
MM000210
MM000220
MM000230
MM000240
MM000250
MM000260
MM000270
MM000280
MM000290
MM000300
MM000310
MM000320
MM000330
MM000340
MM000350
MM000360
MM000370
MM000380
MM000390
MM000400
MM000410
MM000420
MM000430
MM000440

```

SUBROUTINE MMRAN(RAINRT,T,FREQ,RTYPE,GRAIN,BSRAIN)
REF: OLSEN, ET AL., IEEE ANT. PROP., VOL. 26, P. 318(1978)

ROUTINE TO COMPUTE THE ATTENUATION DUE TO RAIN, FOR
FREQUENCIES BETWEEN 10 & 1000 GHZ; BACKSCATTER ALSO.

INPUTS ARE: FREQUENCY(GHZ), TEMPERATURE(KELVIN), RAIN RATE(MM/HR)
OUTPUT IS ATTENUATION (1/KM), BACKSCATTER (MM**2/MM**3)

CALLLED FROM MMWMOD CALLS NO SUBROUTINES.

LOCAL VARIABLES:

ALFA, BETA

BSAT,BSBT

```

A AND B PARAMETERS FUNCTION OF FREQ,
TEMPERATURE, AND RAIN TYPE
POLYNOMIAL COEF.'S FOR LINEAR FIT
TO BACKSCATTER= ALF*RAINRT**BET

```

```
-----  
DIMENSION ALFA(9,3,3), BETA(9,3,3), F(9), TK(3),A(9,3),B(9,3),
```

[illegible][illegible]

```
TR=T
GRAIN=0.
BSRAIN=0.
IF(FREQ.LT.10.) GO TO 200
FR=FREQ
```

```

ITYPE=IFIX(RTYPE+0.1)
DO 10 I=1,3
  DO 10 J=1,9
    A(J,I)=ALFA(J,I,ITYPE)
    B(J,I)=BETA(J,I,ITYPE)
  
```

10 CONTINUÉ

```
CALL INTRP(A,B,TK,F,TR,FR,AA,BB,J)
GRAIN=(AA*RAINRT**BB)/4.343
```

CALCULATIONS FOR RAIN BACKSCATTER.

```
AA=0.  
BB=0.  
IA=2  
IB=2
```

MMR00010
MMR00020
MMR00030
MMR00040
MMR00050
MMR00060
MMR00070
MMR00080
MMR00090
MMR00100
MMR00110
MMR00120
MMR00130
MMR00140
MMR00150
MMR00160
MMR00170
MMR00180
MMR00190
MMR00200
MMR00210
MMR00220
MMR00230
MMR00240
MMR00250
MMR00260
MMR00270
MMR00280
MMR00290
MMR00300
MMR00310
MMR00320
MMR00330
MMR00340
MMR00350
MMR00360
MMR00370
MMR00380
MMR00390
MMR00400
MMR00410
MMR00420
MMR00430
MMR00440
MMR00450
MMR00460
MMR00470
MMR00480
MMR00490
MMR00500
MMR00510
MMR00520
MMR00530
MMR00540
MMR00550
MMR00560
MMR00570
MMR00580
MMR00590
MMR00600
MMR00610
MMR00620
MMR00630
MMR00640
MMR00650
MMR00660
MMR00670
MMR00680
MMR00690
MMR00700

```

IF<FREQ.GT.87.> IA=1
IF<FREQ.GT.82.> IB=1
DO 100 K=1,6
AA=AA+BSAT(K,IA)*(FREQ**(K-1))
BB=BB+BSBT(K,IB)*(FREQ**(K-1))
100 CONTINUE
AA=EXP(AA)
BSRAIN=AA*RAINRT**BB
200 RETURN
END

```

```

MMR00710
MMR00720
MMR00730
MMR00740
MMR00750
MMR00760
MMR00770
MMR00780
MMR00790
MMR00800

```

SUBROUTINE MMSNO(SNRT,TK,FQ,SNEX,SNBS)

PURPOSE: TO COMPUTE SNOW EXTINCTION AND BACKSCATTER X-SECTION.

INPUTS: SNOWFALL RATE(NM/HR), TEMPERATURE(KELVIN), FREQUENCY(GHZ)

OUTPUT: SNOW EXTINCTION (1/KM), BACKSCATTER(M**2/M**3).

CALLED FROM HMMW.

LOCAL VARIABLES:

FF,TS LOCAL FREQUENCY,TEMPERATURE
ITYPE INTEGER SNOW TYPE
XLMOD WAVELENGTH(MM)
FF0,1 FREQUENCY FITTING FACTORS
TF0,1 TEMPERATURE FITTING FACTORS
AP0,1 INTERMEDIATE A VALUES
BP0,1 INTERMEDIATE B VALUES
AA,BB TERMS IN EXT(SNOW)=AA*SNOWRATE**BB
HBSLT TERM IN BSCAT(SNOW)=AA*SNOWRATE**1.8

DATA A(9,3),B(9,3),F(9),T(3),BSAT(6,2),BSBT(6,2),SFCT(3)

DATA A(1,3) 3.0E-3,2.75E-3,1.25E-2,2.50E-2,8.00E-2,1.65E-1,
2.60E-1,5.91E-1,1.68E+0,
2.07E-2,4.34E-2,1.60E-1,2.00E-1,3.10E-1,4.00E-1,
5.81E-1,6.50E-1,1.11E+0,
6.22E-2,9.67E-2,2.35E-1,3.41E-1,6.10E-1,8.52E-1,
7.83E-1,7.37E-1,5.76E-1

DATA B(1,3) 3.146,1.6,1.54,1.26,1.1, .89, .79, .6,

1.3,1.2, .95, .80, .75, .67, .65, .64, .60,

1.3,1.2, .95, .80, .75, .67, .65, .64, .60

DATA F(1,3) 10, 15, 35, 50, 95, 140, 225, 312, 1000

DATA T(1,3) 271, 273, 275, SFCT(1,3) 1, 3, 4

DATA BSAT(1,2) -.8824881E+01, -.1029998E-01, +.2451205E-04,

+ .2462900E-07, +.6507628E-10, -.1856080E-12,

+ .2127020E+02, +.6906017E+00, -.1924260E-01,

+ .3035233E-03, -.2545323E-05, +.8673581E-08,

+ BSBT(1,2) .7901887E+00, -.1900189E-02, +.6341350E-05,

+ .3186429E-08, -.5933950E-10, +.9056715E-13,

+ .1361993E+01, +.3628100E-01, -.2461234E-02,

+ .4805257E-04, -.3988064E-06, +.1193458E-08

TS=TK

FS=FQ

ICE=1

SNEX=0.

SNBS=0.

IF(FQ.LT.10.) GO TO 200

XLMOD=299.79/FQ

CALL INTRP(A,B,T,F,TS,FS,AA,BB,J)

SNEX=AA*SNRT**BB

CALCULATIONS FOR SNOW BACKSCATTER.

AA=0.

BB=0.

IA=2

IB=2

IF(FQ.GT.87.) IA=1

IF(FQ.GT.82.) IB=1

DO 100 K=1,6

AA=AA+BSAT(K,IA)*(FQ**(K-1))

BB=BB+BSBT(K,IB)*(FQ**(K-1))

CONTINUE

AA=EXP(AA)

BB=1.2*BB

F1=(FQ-10.)/85.

FCT=.367+F1*.633

MMS00010
MMS00020
MMS00030
MMS00040
MMS00050
MMS00060
MMS00070
MMS00080
MMS00090
MMS00100
MMS00110
MMS00120
MMS00130
MMS00140
MMS00150
MMS00160
MMS00170
MMS00180
MMS00190
MMS00200
MMS00210
MMS00220
MMS00230
MMS00240
MMS00250
MMS00260
MMS00270
MMS00280
MMS00290
MMS00300
MMS00310
MMS00320
MMS00330
MMS00340
MMS00350
MMS00360
MMS00370
MMS00380
MMS00390
MMS00400
MMS00410
MMS00420
MMS00430
MMS00440
MMS00450
MMS00460
MMS00470
MMS00480
MMS00490
MMS00500
MMS00510
MMS00520
MMS00530
MMS00540
MMS00550
MMS00560
MMS00570
MMS00580
MMS00590
MMS00600
MMS00610
MMS00620
MMS00630
MMS00640
MMS00650
MMS00660
MMS00670
MMS00680
MMS00690
MMS00700

IF(FQ.GT.95.) FCT=1.
AA=FCT*AA
110 IF(TK.GE.275.) J=4
C SNBS=SFCT(J-1)*AA*SNRT**BB
C 200 RETURN
C END

MMS00710
MMS00720
MMS00730
MMS00740
MMS00750
MMS00760
MMS00770
MMS00780

C	SUBROUTINE MMWFG(FD,T,FREQ,GFOG,BSFOG)	MMF00010
C	*****	MMF00020
C	CALCULATES ABSORPTION DUE TO WATER FOGS/CLOUDS,	MMF00030
C	AND BACKSCATTER CROSS SECTION IN M**2/M**3,	MMF00040
C	INPUTS ARE: FOG DENSITY(GM/M**3), TEMPURTURE(KELVIN),	MMF00050
C	FREQUENCY(GHZ).	MMF00060
C	OUTPUTS ARE: FOG ABSORPTION(/KM), BACKSCATTER X-SECTION(M**2 M**3).	MMF00070
C	MMFOG IS CALLED FROM MMWMOD CALLS MMIDX SUBROUTINE.	MMF00080
C	LOCAL VARIABLES:	MMF00090
C	XLMDA WAVELENGTH(MM)	MMF00100
C	*****	MMF00110
C	ICE=0	MMF00120
C	IF(T.LT.243.) ICE=1	MMF00130
C	COMPUTE FOG EXTINCTION	MMF00140
C	XLMDA=10./((FREQ/29.98)	MMF00150
C	CALL MMIDX(XLMDA,T,ICE,H2OAB,H2OK2)	MMF00160
C	GFOG=18.8498*H2OAB*FD/XLMDA	MMF00170
C	BSFOG=1.162E-06*H2OK2*FD**1.75/(XLMDA**4)	MMF00180
C	RETURN	MMF00190
C	END	MMF00200
		MMF00210
		MMF00220
		MMF00230
		MMF00240
		MMF00250
		MMF00260
		MMF00270

```

SUBROUTINE MMWGS(T,P,AH,FREQ,GAS)
*****
SUBROUTINE COMPUTES GASEOUS ABSORPTION FROM 0 TO 1000 GHZ FOR
H2O VAPOR AND O2.
INPUTS INCLUDE: TEMPERATURE(KELVIN), PRESSURE(TORR), ABSOLUTE HUMIDITY
IN GM/M**3, FREQUENCY(GIGAHERTZ).
OUTPUTS ARE: GAS ABSORPTION (1/KM)
MMWGS IS CALLED FROM MMW, CALLS (1)MMOXY(OXYGEN ABSORPTION),
(2) MMH2O(H2O VAPOR ABSORPTION).
LOCAL VARIABLES:
DATA02(L,J): OXYGEN LINE DATA, L=LINE NUMBER, J=TYPE:
      J=1 : LINE FREQUENCY(GHZ)
      2 : LINE STRENGTH AT 300K
      3 : LINE WIDTH AT 300K (GHZ/TORR)
      4 : INTERFERENCE PARAMETER AT 300K
      5 : INTERFERENCE TEMPERATURE CORRECTION
      6 : LINE QUANTUM PARAMETER
DATH20(L,J) : H2O LINE DATA, L=LINE NUMBER, J=TYPE:
      1 : WAVENUMBER(1/CM)
      2 : STRENGTH
      3 : WIDTH(1/CM/TORR)
      4 : GROUND STATE ENERGY
GH20      : H2O VAPOR ABSORPTION (1/KM)
GO2       : O2 ABSORPTION (1/KM)
PH20      : H2O VAPOR PRESSURE(TORR)
WVNMB     : WAVENUMBER(1/CM)
*****
DIMENSION DATA02(42,6), DATH20(37,4)
DIMENSION D102(42,3), D202(42,3), D1H20(37,2), D2H20(37,2)
EQUIVALENCE (DATA02(1,1),D102(1,1)),(DATA02(1,4),D202(1,1))
EQUIVALENCE (DATH20(1,1),D1H20(1,1)),(DATH20(1,3),D2H20(1,1))
COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUM
C*****
DATA D102/
1 49.451,49.961,50.473,50.987,51.503,52.021,52.542,53.066,
2 53.595,54.129,54.671,55.221,55.783,56.264,56.363,56.968,
3 57.612,58.323,58.446,59.164,59.590,60.306,60.434,61.150,
4 61.800,62.411,62.486,62.998,63.568,64.127,64.678,65.224,
5 65.764,66.302,66.836,67.369,67.900,68.430,68.960,69.488,
6 70.016,118.75,
1 7.E-5,2.2E-4,6.E-4,1.56E-3,3.86E-3,8.99E-3,1.971E-2,.04072,
2 .07919,.1448,.2489,.4012,.6056,.3487,.8539,1.1204,1.3595,
3 1.515,.9251,1.5263,1.341,1.3487,1.5626,1.5899,1.4588,1.2272,
4 .9634,.954,.6898,.4656,.2942,.1744,.0971,.0508,.025,.0116,
5 5.08E-3,2.1E-3,8.2E-4,3.E-4,1.E-4,.5973,
1 1.260,1.310,1.330,1.360,1.380,1.410,1.440,1.460,1.490,
2 1.510,1.540,1.570,1.601,2.212,1.635,1.672,1.714,1.762,
3 1.964,1.819,1.859,1.890,1.789,1.736,1.694,1.658,1.990,
4 1.627,1.598,1.568,1.540,1.510,1.490,1.460,1.440,1.410,
5 1.380,1.360,1.330,1.310,1.280,2.140,
DATA D202/
1 0.000,0.000,0.000,1.040,0.802,0.897,0.825,0.780,0.764,
2 0.666,0.651,0.550,0.481,0.931,0.371,0.254,0.100,-.087,
3 0.729,-.318,0.433,-.543,0.179,-.028,-.183,-.324,-.615,
4 -.419,-.537,-.591,-.693,-.703,-.796,-.808,-.849,-.916,
5 -.822,-1.05,0.000,0.000,0.000,-.054,
1 1.00,1.00,1.00,1.38,2.04,1.69,1.91,1.88,1.90,2.01,1.95,
2 2.11,2.13,0.89,2.36,2.66,4.20,-5.8,0.79,0.11,0.50,0.69,
3 -.99,7.60,3.04,2.34,0.85,2.24,2.02,2.04,1.89,1.95,1.85,
4 1.83,1.86,1.66,1.99,1.36,1.00,1.00,1.00,0.89,
1 41.,39.,37.,35.,33.,31.,29.,27.,25.,23.,21.,19.,17.,15.,13.,
2 11.,9.,7.,5.,3.,1.,.13.,.13.,.15.,.17.,.19.,.21.,.23.,.25.,
3 27.,29.,31.,33.,35.,37.,39.,41.,1./
C*****

```

DATA DIH20/	MMG00710
1 00.742,06.115,06.790,10.715,10.846,12.682,14.778,14.944,	MMG00720
2 15.707,15.834,16.294,18.270,18.577,20.704,21.960,24.860,	MMG00730
3 25.085,30.560,32.366,32.954,36.604,37.137,37.910,38.638,	MMG00740
4 38.791,39.112,40.282,40.520,42.639,43.243,43.631,44.100,	MMG00750
5 46.750,47.053,48.059,49.765,49.820,	MMG00760
1 .436E-24,.775E-22,.186E-24,.250E-23,.906E-22,.827E-21,	MMG00770
2 .145E-22,.863E-21,.270E-22,.108E-21,.219E-22,.985E-22,	MMG00780
3 .526E-19,.565E-21,.531E-22,.664E-22,.347E-19,.143E-20,	MMG00790
4 .160E-20,.252E-19,.164E-18,.502E-19,.333E-21,.242E-20,	MMG00800
5 .179E-18,.197E-21,.558E-19,.155E-21,.707E-21,.687E-21,	MMG00810
6 .511E-22,.568E-20,.302E-21,.142E-18,.930E-21,.399E-22,	MMG00820
7 .478E-22/	MMG00830
DATA D2H20/	MMG00840
1 .081,.094,.095,.063,.087,.091,.050,.083,.061,.071,.075,	MMG00850
2 .111,.107,.072,.111,.103,.102,.084,.083,.101,.097,.099,	MMG00860
3 .094,.073,.094,.063,.093,.098,.066,.074,.060,.081,.084,	MMG00870
4 .091,.078,.096,.097,	MMG00880
1 446.512,136.164,134.800,1284.921,315.780,212.156,	MMG00890
2 1045.069,285.419,742.079,488.136,586.482,23.750,	MMG00900
3 23.794,488.110,1618.550,69.920,70.091,285.219,	MMG00910
4 383.843,37.137,136.761,0.000,172.880,610.345,	MMG00920
5 173.366,888.641,275.498,1731.890,888.607,842.361,	MMG00930
6 1079.088,508.814,398.392,399.459,601.553,100.391,	MMG00940
7 1693.650/	MMG00950
C *****	MMG00960
C	MMG00970
C COMPUTE WATER VAPOR PRESSURE, FREQUENCY IN WAVENUMBERS	MMG00980
PH20=AH*T*3.462977E-3	MMG00990
IF(AH.LT.0.) PH20=-PSAT(T)*AH/100.	MMG01000
WVNMB=FRQ/29.98	MMG01010
C	MMG01020
C COMPUTE H2O ABSORPTION	MMG01030
CALL MMH20(WVNMB,T,P,PH20,DATH20,GH20)	MMG01040
C	MMG01050
C COMPUTE O2 ABSORPTION	MMG01060
IF(FRQ.LT.140.) CALL MMOXY(FRQ,T,P,PH20,DATA02,G02)	MMG01070
C	MMG01080
C SUM ABSORPTION	MMG01090
GAS=GH20+G02	MMG01100
C	MMG01110
C *****	MMG01120
RETURN	MMG01130
END	MMG01140

C	FUNCTION PFR(T)	PFR00010
C	COMPUTE H2O PARTITION FUNCTION CORRECTIONS	PFR00020
	DIMENSION VIB(3)	PFR00030
C	DATA VIB/3693.9,1614.5,3801.8/	PFR00040
	QJ=296./T	PFR00050
	QJ=QJ*SQRT(QJ)	PFR00060
	T1=-1.43879/296.	PFR00070
	T2=-1.43879/T	PFR00080
	T1S=1.	PFR00090
	T2S=1.	PFR00100
C	DO 10 J=1,3	PFR00110
	V=VIB(J)	PFR00120
	T11=1.-EXP(T1*V)	PFR00130
	T22=1.-EXP(T2*V)	PFR00140
	T1S=T1S*T11	PFR00150
10	T2S=T2S*T22	PFR00160
	PFR=QJ*T2S/T1S	PFR00170
C	RETURN	PFR00180
	END	PFR00190
		PFR00200
		PFR00210
		PFR00220
		PFR00230

```

C      FUNCTION PSAT(T)
      DATA C1,C2,C3,C4,C5,C6,C7/-7.90298,5.02808,-1.3816E-7,11.344,
+      8.1328E-3,-3.49149,3.005715/
      DATA D1,D2,D3,D4/-9.09718,-3.56654,.876793,.785835/
      DATA TS,TO/373.16,273.16/,CONV/.7500646/
      IF(T.LE.TO) GO TO 100
      TR=TS/T
      TRI=T/TS
      EW=C1*(TR-1.)*C2*ALOG10(TR)+C3*(10.**(C4*(1.-TRI))-1.)*
+      C5*(10.**(C6*(TR-1.))-1.)*C7
      GO TO 200
100    TR=TO/T
      TRI=T/TO
      EW=D1*(TR-1.)*D2*ALOG10(TR)+D3*(1.-TRI)+D4
200    PSAT=(10.**(EW))*CONV
      RETURN
      END

```

```

PSAT0010
PSAT0020
PSAT0030
PSAT0040
PSAT0050
PSAT0060
PSAT0070
PSAT0080
PSAT0090
PSAT0100
PSAT0110
PSAT0120
PSAT0130
PSAT0140
PSAT0150
PSAT0160
PSAT0170
PSAT0180

```

C	FUNCTION SUPK(A,B,C)	SUPK0010
C	COMPUTES THE SUPER KINETIC LINE PROFILE FACTOR.	SUPK0020
	PI=3.14159265	SUPK0030
	XNORM=.998776	SUPK0040
	VM=10.*C	SUPK0050
	X=ABS((B*B-A*A)/(2.*C*A))	SUPK0060
	IF(ABS(B-A).GT.VM) GO TO 10	SUPK0070
C	SUPK=XNORM/(PI*C)/(X*X+1.)	SUPK0080
	RETURN	SUPK0090
C		SUPK0100
10	CXI=10.*(B+10.*C/2.)/(B+10.*C)	SUPK0110
	CXI=((CXI**1.88)+1.)/(CXI*CXI+1.)	SUPK0120
	SUPK=XNORM*CXI/(PI*C)/(X**1.88)+1.)	SUPK0130
	RETURN	SUPK0140
	END	SUPK0150
		SUPK0160
		SUPK0170

```

SUBROUTINE CLTRAN(CTrans,WAVE,IRUN,IERR)
DIMENSION TAU(20)
DIMENSION X1(20),Y1(20),Z1(20),X0(20),Y0(20),Z0(20)
COMMON /CLYMAT/TEMP,PRESS,RH,AH,DP,VIS,CLDAMT,CLDHYT,FOGPRB,
+ WNDVEL,WNDDIR,IPASCT
COMMON /LEVEL/ISLTUP,ISLTDN,IHORIZ,IVERT,TESALL,TESARG,TESVT
COMMON /PATHL/X0,Y0,Z0,X1,Y1,Z1,XS,YS,ZS,XT,YT,ZT,ATTIL(20)
COMMON /BASTOP/ZBAS,ZTOP,ICL,IWV,ILCTYP(10),ICCTYP(10)
COMMON /CYL/XC,YC,RADIUS
COMMON /IQUINT/IQIN,IQOUT,IPHFUN,LOUNIT,NDIRTU,NCLINT,KSTOR,HPLCTU
COMMON /BASTH/ZLBASE(10),ZLTHIC(10),ZCBASE(10),ZCTHIC(10)
+ ,RADICL(10)
COMMON /INTCL/XCLOUD(10),YCLOUD(10),NLINT(10),NCINT(10)
C** DATA IS READ FROM INPUT RECORDS AND THEN TRANSFERRED TO CLTRAN
C** BY SUBROUTINE CLREAD. INPUT TO CLTRAN IS CARD ORDER-INDEPENDENT,
C** WITH A FOUR-LETTER IDENTIFIER IN COLUMNS 1-4 OF EACH INPUT RECORD.
C** THE ONLY EXCEPTION TO THIS ORDER-INDEPENDENCE IS THE GO SENTINEL
C** CARD, WHICH MUST BE THE LAST RECORD READ. ALL CARDS ARE READ IN
C** UNDER THE FORMAT (A4,1X,5(E10.5,1X)), THE IDENTIFICATION AND MEANING
C** OF EACH INPUT RECORD ARE AS FOLLOWS :
C-----
C CARD IDENTIFIER : SEEK
C VARIABLES READ : XS,YS,ZS
C XS,YS,ZS = POSITION COORDINATES OF SEEKER (OR RECEIVER) (KM)
C-----
C CARD IDENTIFIER : TARG
C VARIABLES READ : XT,YT,ZT
C XT,YT,ZT = POSITION COORDINATES OF TARGET (KM)
C-----
C** THE FOLLOWING FOUR CARDS REPRESENT THE STRATIFORM CLOUDS TREATED
C** BY CLTRAN. THEY ARE TREATED AS INFINITE LAYERS IN A PLANE-PARALLEL
C** ATMOSPHERE.
C-----
C CARD IDENTIFIER : CLST (STRATUS CLOUD TYPE)
C VARIABLES READ : ZLBASE,ZLTHIC
C ZLBASE = HEIGHT OF CLOUD BASE (KM)
C ZLTHIC = VERTICAL THICKNESS OF CLOUD LAYER (KM)
C-----
C CARD IDENTIFIER : CLAS (ALTOSTRATUS CLOUD TYPE)
C VARIABLES READ : ZLBASE,ZLTHIC
C-----
C CARD IDENTIFIER : CLNS (NIMBOSTRATUS CLOUD TYPE)
C VARIABLES READ : ZLBASE,ZLTHIC
C-----
C CARD IDENTIFIER : CLSC (STRATOCUMULUS CLOUD TYPE)
C VARIABLES READ : ZLBASE,ZLTHIC
C-----
C** THE NEXT TWO CARDS REPRESENT THE CUMULUS CLOUD TYPES ADDRESSED
C** BY CLTRAN. CLOUDS OF THIS KIND ARE MODELLED AS CYLINDERS WHICH
C** HAVE VERTICAL SYMMETRY AXES. THESE TYPES ARE REPRESENTED BY
C** THE FOLLOWING RECORDS :
C-----
C CARD IDENTIFIER : CLCH (CUMULUS HUMILIS CLOUD TYPE)
C VARIABLES READ : ZCBASE,ZCTHIC,RADICL,XCLOUD,YCLOUD
C ZCBASE = HEIGHT OF CLOUD CYLINDER'S LOWER BASE (KM)
C ZCTHIC = VERTICAL THICKNESS OF CLOUD CYLINDER (KM)
C RADICL = RADIUS OF CLOUD CYLINDER (KM)
C XCLOUD = X-COORDINATE OF VERTICAL AXIS OF CLOUD CYLINDER (KM)
C YCLOUD = Y-COORDINATE OF VERTICAL AXIS OF CLOUD CYLINDER (KM)
C-----
C CARD IDENTIFIER : CLCC (CUMULUS CONGESTUS CLOUD TYPE)
C VARIABLES READ : ZCBASE,ZCTHIC,RADICL,XCLOUD,YCLOUD
C-----
C** THE FOLLOWING CARD MUST BE THE LAST RECORD READ :

```

C		CLT00690
C		CLT00700
C	CARD IDENTIFIER : GO	CLT00710
C	VARIABLES READ : NONE	CLT00720
C		CLT00730
	IF<IRUN.GT.1> GO TO 5	CLT00740
	NLAYER=0	CLT00750
	NCCLDS=0	CLT00760
5	CONTINUE	CLT00770
	IWV=0	CLT00780
	ZBMIN=999.	
	FOGPRB=0.	
	IF<WAVE.GT.0.20.AND.WAVE.LT.2.00> IWV=1	CLT00790
	IF<WAVE.GT.3.00.AND.WAVE.LT.5.00> IWV=2	CLT00800
	IF<WAVE.GT.8.0.AND.WAVE.LT.12.0> IWV=3	CLT00810
	IF<IWV.NE.0> GO TO 50	CLT00820
	IERR=1	CLT00830
	WRITE<IOOUT,22>	CLT00840
22	FORMAT<1H0,20X,94H***CLTRAN ERROR*** INPUT WAVELENGTH DOES NOT LIE	CLT00850
	+ WITHIN ALLOWABLE LIMITS, EXECUTION TERMINATED />	CLT00860
	GO TO 900	CLT00870
50	CALL CLREAD<NLAYER,NCCLDS,IERR>	CLT00880
	IF<IERR.EQ.1> GO TO 900	CLT00890
	ISLTUP=0	CLT00900
	ISLTDN=0	CLT00910
	IHORIZ=0	CLT00920
	IVERT=0	CLT00930
C	C**** DETERMINE SENSE OF L-O-S SLOPE FROM SEEKER'S POINT OF VIEW	CLT00940
C		CLT00950
	TESVER=(XS-XT)**2+(YS-YT)**2	CLT00960
	IF<TESVER.EQ.0.0>IVERT=1	CLT00970
	IF<IVERT.EQ.1>GO TO 300	CLT00980
	IF<ZS-ZT>200,210,220	CLT00990
200	ISLTUP=1	CLT01000
	GO TO 300	CLT01010
210	IHORIZ=1	CLT01020
	GO TO 300	CLT01030
220	ISLTDN=1	CLT01040
300	CONTINUE	CLT01050
	IF<IHORIZ.EQ.1>GO TO 310	CLT01060
C	C**** COMPUTE L-O-S SLOPES IN X-Z AND Y-Z VERTICAL PLANES	CLT01070
C		CLT01080
	XIX=(XS-XT)/(ZS-ZT)	CLT01090
	XIY=(YS-YT)/(ZS-ZT)	CLT01100
310	CONTINUE	CLT01110
C	C**** STRATIFORM CLOUD BLOCK	CLT01120
C		CLT01130
	IF<NLAYER.EQ.0>GO TO 500	CLT01140
C	C**** UTILIZE DEFAULT BASE OR THICKNESS VALUES IF NECESSARY	CLT01150
C		CLT01160
	CALL DEFSET<1,NLAYER>	CLT01170
	DO 400 N=1,NLAYER	CLT01180
	NLINT(N)=0	CLT01190
	Z0(N)=ZLBASE(N)	CLT01200
	Z1(N)=Z0(N)+ZLTHIC(N)	CLT01210
	ZBAS=Z0(N)	CLT01220
	ZTOP=Z1(N)	CLT01230
	ICL=ILCTYP(N)	CLT01240
	IF<ZBMIN.LT.ZLBASE(N)> GO TO 320	CLT01250
	ZBMIN=ZLBASE(N)	CLT01260
	FOGPRB=FLOAT<ICL>	CLT01270
320	CONTINUE	CLT01280
C		CLT01290
C****	DETERMINE X,Y,Z INTERSECTIONS OF L-O-S AND CLOUD LAYER <IF	CLT01300
C****	THERE ARE ANY>: <X1(N),Y1(N),Z1(N)>= UPPER INTERSECTION POINT,	CLT01310
C****	<X0(N),Y0(N),Z0(N)>= LOWER INTERSECTION POINT	CLT01320


```

C      CALL LAYRXY(XIX,XIY,N)
C      IF(TESALL.LE.0.0)GO TO 350
C      NLINT(N)=N
C**** IF THERE ARE ANY INTERSECTIONS, DETERMINE OPTICAL DEPTH
C      CALL CLEXTN(TAUN,N)
C      TAU(N)=TAUN
C      ATTL(N)=SQRT((X1(N)-X0(N))**2+(Y1(N)-Y0(N))**2+(Z1(N)-Z0(N))**2)
C      GO TO 400
C**** IF NO INTERSECTIONS WERE FOUND, THE OPTICAL DEPTH
C**** IS NOW SET TO ZERO
C      350 TAU(N)=0.0
C      ATTL(N)=0.0
C      400 CONTINUE
C**** END STRATIFORM CLOUD BLOCK
C**** CUMULIFORM CLOUD BLOCK
C      500 IF(NCCLDS.EQ.0)GO TO 805
C**** UTILIZE DEFAULT BASE, THICKNESS, OR RADIUS IF NECESSARY
C      CALL DEFSET(2,NCCLDS)
C      DO 800 N=1,NCCLDS
C      NN=N+1
C      NCINT(N)=0
C      Z0(NN)=ZCBASE(N)
C      Z1(NN)=Z0(NN)+ZCTHIC(N)
C      ICL=ICCTYP(N)
C      IF(ZBMIN.LT.ZCBASE(N)) GO TO 520
C      ZBMIN=ZCBASE(N)
C      FOGPRB=FLOAT(ICL)
C      520 CONTINUE
C      RADIUS=RADICL(N)
C      XC=XCLOUD(N)
C      YC=YCLOUD(N)
C      ZBAS=Z0(NN)
C      ZTOP=Z1(NN)
C      TESARG=0.0
C      TESVT=0.0
C**** DETERMINE X,Y,Z INTERSECTIONS OF L-O-S AND CUMULUS CLOUD CYLINDER
C**** (IF THERE ARE ANY): (X1(N),Y1(N),Z1(N))= UPPER INTERSECTION POINT,
C**** (X0(N),Y0(N),Z0(N))= LOWER INTERSECTION POINT
C      CALL CYLXY(XIX,XIY,NN)
C      IF(TESALL.LE.0.0.OR.TESARG.LT.0.0.OR.TESVT.LT.0.0)GO TO 650
C      NCINT(N)=N
C**** IF THERE ARE ANY INTERSECTIONS, DETERMINE OPTICAL DEPTH
C      CALL CLEXTN(TAUN,NN)
C      TAU(NN)=TAUN
C      ATTL(NN)=SQRT((X1(NN)-X0(NN))**2+(Y1(NN)-Y0(NN))**2+
C      +(Z1(NN)-Z0(NN))**2)
C      GO TO 800
C**** IF NO INTERSECTIONS WERE FOUND, THE OPTICAL DEPTH
C**** IS NOW SET TO ZERO
C      650 TAU(NN)=0.0
C      ATTL(NN)=0.0
C      800 CONTINUE
C**** END CUMULIFORM CLOUD BLOCK

```

```

CLT01330
CLT01340
CLT01350
CLT01360
CLT01370
CLT01380
CLT01390
CLT01400
CLT01410
CLT01420
CLT01430
CLT01440
CLT01450
CLT01460
CLT01470
CLT01480
CLT01490
CLT01500
CLT01510
CLT01520
CLT01530
CLT01540
CLT01550
CLT01560
CLT01570
CLT01580
CLT01590
CLT01600
CLT01610
CLT01620
CLT01630
CLT01640
CLT01650
CLT01660
CLT01670
CLT01680
CLT01690
CLT01700
CLT01710
CLT01720
CLT01730
CLT01740
CLT01750
CLT01760
CLT01770
CLT01780
CLT01790
CLT01800
CLT01810
CLT01820
CLT01830
CLT01840
CLT01850
CLT01860
CLT01870
CLT01880
CLT01890
CLT01900
CLT01910
CLT01920
CLT01930
CLT01940
CLT01950
CLT01960
CLT01970
CLT01980

```

```

C
805 CONTINUE
C**** DETERMINE CUMULATIVE CLOUD OPTICAL DEPTH AND TRANSMITTANCE
      TAUTOT=0.0
      NN=Nlayer+NCCLDS
      IF(NN.EQ.0)GO TO 820
      DO 810 N=1,NN
810   TAUTOT=TAUTOT+TAU(N)
820   CTRANS=EXP(-TAUTOT)
850   CALL LISOUT(NN,CTrans,TAUTOT,TAU,Nlayer,IRUN)
900   RETURN
      END

```

```

CLT01990
CLT02000
CLT02010
CLT02020
CLT02030
CLT02040
CLT02050
CLT02060
CLT02070
CLT02080
CLT02090
CLT02100

```

SUBROUTINE LAYRXY(XIX,XIY,IN)	LAY00010
LOGICAL INSEEK,INTARG	LAY00020
DIMENSION X1(20),Y1(20),Z1(20),X0(20),Y0(20),Z0(20)	LAY00030
COMMON /LEVEL/ISLTUP,ISLTDN,IHORIZ,IVERT,TESALL,TESARG,TESVT	LAY00040
COMMON /PATHL/X0,Y0,Z0,X1,Y1,Z1,XS,YS,ZS,XT,YT,ZT,ATTIL(20)	LAY00050
C**** CHECK LOS TO SEE IF IT INTERSECTS THE CLOUD LAYER	LAY00060
UP1=ISLTUP*(ZS-Z1(IN))	LAY00070
UP2=ISLTUP*(Z0(IN)-ZT)	LAY00080
DN1=ISLTDN*(ZT-Z1(IN))	LAY00090
DN2=ISLTDN*(Z0(IN)-ZS)	LAY00100
HOR12=IHORIZ*(ZS-Z1(IN))*(ZS-Z0(IN))*(-1,0)	LAY00110
UPVER1=IVERT*(ZT-ZS)*(ZS-Z1(IN))	LAY00120
UPVER2=IVERT*(ZT-ZS)*(Z0(IN)-ZT)	LAY00130
TESALL=UP1*UP2+DN1*DN2+HOR12+UPVER1*UPVER2	LAY00140
IF(TESALL.LT.0.0)RETURN	LAY00150
IF(IHORIZ.EQ.1)RETURN	LAY00160
C	LAY00170
C**** COMPUTE X,Y INTERSECTIONS OF CLOUD PLANES AND LOS	LAY00180
C**** ALSO, CHECK FOR THE CASE WHERE EITHER THE SEEKER	LAY00190
C**** OR TARGET IS INSIDE OF THE CLOUD	LAY00200
C	LAY00210
INSEEK=ZS.LE.Z1(IN).AND.ZS.GE.Z0(IN)	LAY00220
INTARG=ZT.LE.Z1(IN).AND.ZT.GE.Z0(IN)	LAY00230
IF(ISLTUP.EQ.1.AND.INTARG)Z1(IN)=ZT	LAY00240
IF(ISLTDN.EQ.1.AND.INTARG)Z0(IN)=ZT	LAY00250
IF(ISLTUP.EQ.1.AND.INSEEK)Z0(IN)=ZS	LAY00260
IF(ISLTDN.EQ.1.AND.INSEEK)Z1(IN)=ZS	LAY00270
X1(IN)=XS+XIX*(Z1(IN)-ZS)	LAY00280
Y1(IN)=YS+XIY*(Z1(IN)-ZS)	LAY00290
X0(IN)=XS+XIX*(Z0(IN)-ZS)	LAY00300
Y0(IN)=YS+XIY*(Z0(IN)-ZS)	LAY00310
RETURN	LAY00320
END	LAY00330

```

SUBROUTINE CYLXY(XIX,XIY,IN)
DIMENSION X1(20),Y1(20),Z1(20),X0(20),Y0(20),Z0(20)
COMMON /CYL/XC,YC,RADIUS
COMMON /LEVEL/ISLTUP,ISLTDN,IHORIZ,IVERT,TESALL,TESARG,TESVT
COMMON /PATHL/X0,Y0,Z0,X1,Y1,Z1,XS,YS,ZS,XT,YT,ZT,ATTL(20)
COMMON /BASTOP/ZBAS,ZTOP,ICL,IWV,ILCTYP(10),ICCTYP(10)
C**** FIRST DETERMINE INTERSECTION PTS OF LOS WITH UPPER
C**** AND LOWER BASE PLANES OF CLOUD CYLINDER
TESALL=0.0
CALL LAYRXY(XIX,XIY,IN)
IF(TESALL.LT.0.0)RETURN
X0LAYR=X0(IN)
Y0LAYR=Y0(IN)
Z0LAYR=Z0(IN)
X1LAYR=X1(IN)
Y1LAYR=Y1(IN)
Z1LAYR=Z1(IN)
C**** CALCULATE NEXT THE INTERSECTION PTS OF THE LOS
C**** WITH THE SURFACE OF AN INFINITE VERTICAL CYLINDER WITH
C**** THE SAME RADIUS (RADIUS) AND LATERAL POSITION (XC,YC) AS THE CLOUD
IF(IVERT.NE.1)GO TO 40
VERTES=SQRT((XT-XC)**2+(YT-YC)**2)
TESVT=RADIUS-VERTES
IF(VERTES.LT.RADIUS)GO TO 400
RETURN
40 IF((XT-XS).EQ.0.0)GO TO 60
ALPHA=(YT-YS)/(XT-XS)
A2=ALPHA**2
C2=1.0+A2
C1=2.0*(ALPHA*(YT-YC-ALPHA*XT)-XC)
C0=XC**2+(YT-YC-ALPHA*XT)**2-RADIUS**2
TESARG=C1**2-4.0*C2*C0
IF(TESARG.LT.0.0)RETURN
XP=(-C1+SQRT(TESARG))/(2.0*C2)
YM=(-C1-SQRT(TESARG))/(2.0*C2)
YP=YT+ALPHA*(XP-XT)
YM=YT+ALPHA*(XM-XT)
IF(IHORIZ.EQ.1)GO TO 300
ZM=ZS+(1.0/XIX)*(XM-XS)
ZP=ZS+(1.0/XIX)*(XP-XS)
C**** CHECK FOR SKEW MISS OF CLOUD
TOPP=ZP-ZTOP
TOPM=ZM-ZTOP
BASP=ZBAS-ZP
BASM=ZBAS-ZM
IF((TOPP*TOPM).LT.0.0.OR.(BASP*BASM).LT.0.0)GO TO 50
TESALL=-1.0
RETURN
60 XP=XT
TESARG=RADIUS**2-(XP-XT)**2
IF(TESARG.LT.0.0)RETURN
YP=YC+SQRT(TESARG)
XM=XT
YM=YC+SQRT(TESARG)
IF(IHORIZ.EQ.1)GO TO 300
ZM=ZS+(1.0/XIY)*(YM-YS)
ZP=ZS+(1.0/XIY)*(YP-YS)
C**** CHECK FOR SKEW MISS OF CLOUD
TOPP=ZP-ZTOP
TOPM=ZM-ZTOP
BASP=ZBAS-ZP
BASM=ZBAS-ZM
IF((TOPP*TOPM).LT.0.0.OR.(BASP*BASM).LT.0.0)GO TO 50
TESALL=-1.0
RETURN
50 IF(ZP.LT.ZM)GO TO 100
X0CYL=XM
Y0CYL=YM
Z0CYL=ZM
X1CYL=XP

```

CYL00010
CYL00020
CYL00030
CYL00040
CYL00050
CYL00060
CYL00070
CYL00080
CYL00090
CYL00100
CYL00110
CYL00120
CYL00130
CYL00140
CYL00150
CYL00160
CYL00170
CYL00180
CYL00190
CYL00200
CYL00210
CYL00220
CYL00230
CYL00240
CYL00250
CYL00260
CYL00270
CYL00280
CYL00290
CYL00300
CYL00310
CYL00320
CYL00330
CYL00340
CYL00350
CYL00360
CYL00370
CYL00380
CYL00390
CYL00400
CYL00410
CYL00420
CYL00430
CYL00440
CYL00450
CYL00460
CYL00470
CYL00480
CYL00490
CYL00500
CYL00510
CYL00520
CYL00530
CYL00540
CYL00550
CYL00560
CYL00570
CYL00580
CYL00590
CYL00600
CYL00610
CYL00620
CYL00630
CYL00640
CYL00650
CYL00660
CYL00670
CYL00680
CYL00690
CYL00700

```

      Y1CYL=YP
      Z1CYL=ZP
100   GO TO 200
      X0CYL=XP
      Y0CYL=YP
      Z0CYL=ZP
      X1CYL=XM
      Y1CYL=YM
      Z1CYL=ZM
200   CONTINUE
      GO TO 450
300   X0(IN)=XM
      Y0(IN)=YM
      Z0(IN)=ZT
      X1(IN)=XP
      Y1(IN)=YP
      Z1(IN)=ZT
      GO TO 500
400   X0(IN)=XT
      Y0(IN)=YT
      X1(IN)=XT
      Y1(IN)=YT
      GO TO 500
450   IF(Z0CYL.LT.Z0LAYR)GO TO 460
      X0(IN)=X0CYL
      Y0(IN)=Y0CYL
      Z0(IN)=Z0CYL
      GO TO 470
460   X0(IN)=X0LAYR
      Y0(IN)=Y0LAYR
      Z0(IN)=Z0LAYR
470   IF(Z1CYL.GT.Z1LAYR)GO TO 480
      X1(IN)=X1CYL
      Y1(IN)=Y1CYL
      Z1(IN)=Z1CYL
      GO TO 500
480   X1(IN)=X1LAYR
      Y1(IN)=Y1LAYR
      Z1(IN)=Z1LAYR
500   RETURN
      END

```

```

CYL00710
CYL00720
CYL00730
CYL00740
CYL00750
CYL00760
CYL00770
CYL00780
CYL00790
CYL00800
CYL00810
CYL00820
CYL00830
CYL00840
CYL00850
CYL00860
CYL00870
CYL00880
CYL00890
CYL00900
CYL00910
CYL00920
CYL00930
CYL00940
CYL00950
CYL00960
CYL00970
CYL00980
CYL00990
CYL01000
CYL01010
CYL01020
CYL01030
CYL01040
CYL01050
CYL01060
CYL01070
CYL01080
CYL01090
CYL01100
CYL01110

```

```

SUBROUTINE CLEXTN(TAUN,IN)
DIMENSION AA(42),BB(42),CC(42),A(7,6,3)
DIMENSION X1(20),Y1(20),Z1(20),X0(20),Y0(20),Z0(20)
EQUIVALENCE (AA(1),A(1,1,1)),(BB(1),A(1,1,2)),(CC(1),A(1,1,3))
COMMON /PATHL/X0,Y0,Z0,X1,Y1,Z1,XS,YS,ZS,XT,YT,ZT,ATTL(20)
COMMON /LEVEL/ISLTUP,ISLTDN,IHORIZ,IVERT,TESALL,TESARG,TESVT
COMMON /BASTOP/ZBAS,ZTOP,ICL,IWV
DATA AA/5.3744E-2,2.9313E-4,-1.1300E-6,1.6228E-9,-1.0421E-12,
+2.4192E-16,0.0000E-00,8.4451E-2,1.6549E-4,-4.7300E-7,
+4.4334E-10,-1.3851E-13,-2.0644E-17,1.2569E-20,1.0240E-1,
+4.8678E-5,-2.2195E-7,1.4506E-10,5.9393E-14,-8.7250E-17,
+2.1383E-20,3.6775E-2,1.6160E-4,-6.4666E-7,9.5298E-10,
+-6.4023E-13,1.6916E-16,-7.9924E-21,1.8410E-2,6.6870E-4,
+-2.8406E-6,5.4478E-9,-5.4854E-12,2.7932E-15,-5.8271E-19,
+3.3458E-2,1.3098E-4,-4.1528E-7,6.1166E-10,-4.6681E-13,
+1.7383E-16,-2.5043E-20/
DATA BB/7.5099E-2,3.2061E-4,-1.7060E-6,3.3538E-9,-3.2719E-12,
+1.5721E-15,-3.0123E-19,1.1808E-1,2.2387E-4,-8.7996E-7,
+1.1298E-9,-6.6823E-13,1.7514E-16,-1.5663E-20,1.4155E-1,
+-5.0592E-5,-1.2280E-7,1.2715E-10,2.7470E-14,-6.6940E-17,
+1.7734E-20,4.9533E-2,2.0904E-4,-1.1626E-6,2.3531E-9,
+-2.3463E-12,1.1472E-15,-2.2267E-19,3.8315E-3,1.1837E-3,
+-5.1096E-6,9.4660E-9,-8.6426E-12,3.6778E-15,-5.5775E-19,
+4.1534E-2,2.0220E-4,-8.1463E-7,1.3590E-9,-1.1142E-12,
+4.3867E-16,-6.6426E-20/
DATA CC/1.1269E-2,2.8659E-4,-6.0210E-8,-1.5274E-9,2.7747E-12,
+-1.8946E-15,4.5539E-19,1.9856E-2,1.2292E-4,-8.4479E-8,
+8.6563E-12,0.0000E-0,0.0000E-0,0.0000E-0,3.1907E-2,
+1.9632E-4,-1.1114E-7,-3.3987E-10,5.2528E-13,-2.7799E-16,
+5.0469E-20,6.8522E-3,1.5362E-4,7.5813E-8,-1.1430E-9,
+1.8885E-12,-1.2477E-15,2.9522E-19,8.5792E-4,6.4122E-5,
+7.7271E-7,-2.9750E-9,4.4014E-12,-3.0626E-15,8.1541E-19,
+3.7151E-3,1.4919E-4,-9.3486E-8,-1.2183E-10,1.8733E-13,
+-9.2167E-17,1.5939E-20/
TAUN=0.0
ZZA=Z0(IN)-ZBAS
ZZB=Z1(IN)-ZBAS
VERDIS=ZZB-ZZA
HORDIS=SQRT((X1(IN)-X0(IN))**2+(Y1(IN)-Y0(IN))**2)
EL=SQRT(HORDIS**2+VERDIS**2)
IF(EL.EQ.0.0)RETURN
IF(IHORIZ.EQ.1)GO TO 200
XI=VERDIS/EL
ELA=ZZA*1000.0
ELB=(XI*EL+ZZA)*1000.0
POLYA=0.0
POLYB=0.0
DO 100 N=1,7
EN=FLOAT(N)
AN=A(N,ICL,IWV)
TERMB=(1.0/(XI*EN))*AN*ELB**N
TERMA=(1.0/(XI*EN))*AN*ELA**N
POLYA=POLYA+TERMA
POLYB=POLYB+TERMB
100 TAUN=POLYB-POLYA
GO TO 300
200 ZZA=ZZA*1000.0
DO 250 N=1,7
TERMK=A(N,ICL,IWV)*HORDIS*1000.0*ZZA**(N-1)
250 TAUN=TAUN+TERMK
300 RETURN
END

```

```

CLE00010
CLE00020
CLE00030
CLE00040
CLE00050
CLE00060
CLE00070
CLE00080
CLE00090
CLE00100
CLE00110
CLE00120
CLE00130
CLE00140
CLE00150
CLE00160
CLE00170
CLE00180
CLE00190
CLE00200
CLE00210
CLE00220
CLE00230
CLE00240
CLE00250
CLE00260
CLE00270
CLE00280
CLE00290
CLE00300
CLE00310
CLE00320
CLE00330
CLE00340
CLE00350
CLE00360
CLE00370
CLE00380
CLE00390
CLE00400
CLE00410
CLE00420
CLE00430
CLE00440
CLE00450
CLE00460
CLE00470
CLE00480
CLE00490
CLE00500
CLE00510
CLE00520
CLE00530
CLE00540
CLE00550
CLE00560
CLE00570
CLE00580
CLE00590
CLE00600
CLE00610
CLE00620

```

```

SUBROUTINE LISOUT(NN,CTrans,TAUTOT,TAU,NLAYER,IRUN)
C**** OUTPUT CONTROL ROUTINE
C
DIMENSION ALPH(6),T(20),TAU(20)
DIMENSION X1(20),Y1(20),Z1(20),X0(20),Y0(20),Z0(20)
COMMON /PATHL/ X0,Y0,Z0,X1,Y1,Z1,XS,YS,ZS,XT,YT,ZT,ATT(20)
COMMON /LEVEL/ ISLTUP,ISLTDN,IHORIZ,IVERT,TESALL,TESARG,TESVT
COMMON /BASTOP/ ZBAS,ZTOP,ICL,IWV,ILCTYP(10),ICCTYP(10)
COMMON /BASTH/ ZLBASE(10),ZLTHIC(10),ZCBASE(10),ZCTHIC(10)
+ ,RADICL(10)
COMMON /INTCL/ XCLOUD(10),YCLOUD(10),NLINT(10),NCINT(10)
COMMON /IOUNIT/ IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLINT,KSTOR,NPLOTUL
DATA ALPH/2HST,2HAS,2HNS,2HSC,2HCH,2HCC/
IF(IWV.EQ.1)WVL=0.55
IF(IWV.EQ.2)WVL=3.80
IF(IWV.EQ.3)WVL=10.60
IF(IRUN.LT.2) WRITE(IOOUT,150)WVL
IF(IRUN.GT.1) WRITE(IOOUT,160)WVL
150 FORMAT(1H0,40X,13HWAVELENGTH = ,F5.2,9H MICRONS //)
160 FORMAT(1H0,40X,13HWAVELENGTH = ,F5.2,9H MICRONS //)
IF(IHORIZ.EQ.1)WRITE(IOOUT,200)
IF(IVERT.EQ.1)WRITE(IOOUT,210)
IF(ISLTUP.EQ.1)WRITE(IOOUT,220)
IF(ISLTDN.EQ.1)WRITE(IOOUT,230)
200 FORMAT(1H0,40X,28HLINE-OF-SIGHT IS HORIZONTAL //)
210 FORMAT(1H0,40X,26HLINE-OF-SIGHT IS VERTICAL //)
220 FORMAT(1H0,40X,28HLINE-OF-SIGHT SLANTS UPWARD //)
230 FORMAT(1H0,40X,30HLINE-OF-SIGHT SLANTS DOWNWARD //)
PTHLEN=SQRT((XS-XT)**2+(YS-YT)**2+(ZS-ZT)**2)
WRITE(IOOUT,300)PTHLEN
300 FORMAT(1H0,40X,29HTOTAL LINE-OF-SIGHT LENGTH = ,F7.3,4H KM )
ATTLEN=0.0
IF(TAUTOT.EQ.0.0)GO TO 780
DO 310 N=1,NN
310 ATTLEN=ATTLEN+ATT(N)
WRITE(IOOUT,400)ATTLEN
400 FORMAT(1H0,40X,50HTOTAL LINE-OF-SIGHT LENGTH INTERRUPTED BY CLOUD
+ = ,F7.3,4H KM )
WRITE(IOOUT,500)TAUTOT
500 FORMAT(1H0,40X,22HTOTAL OPTICAL DEPTH = ,F7.2//)
WRITE(IOOUT,600)CTrans
600 FORMAT(1H0,40X,36HTRANSMITTANCE ALONG LINE-OF-SIGHT = ,E11.5,//)
WRITE(IOOUT,605)
605 FORMAT(1H0,40X,50HSEEKER COORDINATES (KM)   TARGET COORDINATES (KM)
+ )
WRITE(IOOUT,606)
606 FORMAT(1H ,40X,23HXSEEKER YSEEKER ZSEEKER,3X,23HXTARGET YTARGET ZT
+ARGET)
WRITE(IOOUT,607)
607 FORMAT(1H ,40X,3(8H----- ),2X,3(8H----- ))
WRITE(IOOUT,609)XS,YS,ZS,XT,YT,ZT
609 FORMAT(1H0,40X,3(F6.3,2X),2X,3(F6.3,2X))
WRITE(IOOUT,613)
613 FORMAT(1H ,//)
WRITE(IOOUT,615)
615 FORMAT(1H0,40X,62HCLLOUD TYPE      LINE-OF-SIGHT INTERSECTION COORD
+ INATES (KM) )
WRITE(IOOUT,620)
620 FORMAT(1H ,40X,10H/ID NUMBER,4X,47HXUPPER YUPPER ZUPPER XLOWER
+ YLOWER ZLOWER)
WRITE(IOOUT,625)
625 FORMAT(1H ,40X,10H-----,4X,3(8H----- ),1X,3(8H----- ))
IF(NLAYER.EQ.0)GO TO 685
DO 670 N=1,NLAYER
IND=ILCTYP(N)
IF(NLINT(N).EQ.0)GO TO 670
WRITE(IOOUT,680)ALPH(IND),N,X1(N),Y1(N),Z1(N),X0(N),Y0(N),Z0(N)
670 CONTINUE
680 FORMAT(1H0,42X,A2,1H/,12,6X,3(F7.3,1X),1X,3(F7.3,1X))

```

685	NLO=Nlayer+1	LIS00710
	IF(Nlayer.EQ.NN)GO TO 699	LIS00720
	LL=0	LIS00730
	DO 690 N=NLO,NN	LIS00740
	LL=LL+1	LIS00750
	IND=ICCTYP(LL)	LIS00760
	IF(NCINT(LL).EQ.0)GO TO 690	LIS00770
	WRITE(IOOUT,680)ALPH(IND),N,X1(N),Y1(N),Z1(N),X0(N),Y0(N),Z0(N)	LIS00780
690	CONTINUE	LIS00790
	WRITE(IOOUT,613)	LIS00800
	WRITE(IOOUT,691)	LIS00810
691	FORMAT(1H0,40X,59H CLOUD TYPE CUMULUS CENTER POSITIONS (FOR CC OR	LIS00820
	+ CH TYPES))	LIS00830
	WRITE(IOOUT,692)	LIS00840
692	FORMAT(1H ,40X,10H/ID NUMBER,6X,26HX CLOUD (KM) Y CLOUD (KM))	LIS00850
	WRITE(IOOUT,693)	LIS00860
693	FORMAT(1H ,40X,10H-----,6X,26H-----)	LIS00870
	LL=0	LIS00880
	DO 695 N=NLO,NN	LIS00890
	LL=LL+1	LIS00900
	IND=ICCTYP(LL)	LIS00910
695	WRITE(IOOUT,696)ALPH(IND),N,X CLOUD(LL),Y CLOUD(LL)	LIS00920
696	FORMAT(1H0,42X,A2,1H/,12,10X,F7.3,8X,F7.3)	LIS00930
699	CONTINUE	LIS00940
	WRITE(IOOUT,613)	LIS00950
	DO 610 N=1,NN	LIS00960
610	T(N)=EXP(-TAU(N))	LIS00970
	IF(NCCLDS.GT.0) WRITE(IOOUT,700)	LIS00980
	IF(NCCLDS.LT.1) WRITE(IOOUT,705)	LIS00990
700	FORMAT(1H1,20X,11H CLOUD TYPE ,2X,15H HEIGHT OF BASE ,2X,	LIS01000
	+10H THICKNESS ,2X,16H RADIUS OF CLOUD ,2X,14H OPTICAL DEPTH ,2X,	LIS01010
	+14H TRANSMITTANCE)	LIS01020
705	FORMAT(1H0,20X,11H CLOUD TYPE ,2X,15H HEIGHT OF BASE ,2X,	LIS01030
	+10H THICKNESS ,2X,16H RADIUS OF CLOUD ,2X,14H OPTICAL DEPTH ,2X,	LIS01040
	+14H TRANSMITTANCE)	LIS01050
	WRITE(IOOUT,710)	LIS01060
710	FORMAT(1H ,20X,10H/ID NUMBER,9X,4H(KM),9X,4H(KM),12X,4H(KM),9X,	LIS01070
	+11HALONG L-O-S,5X,11HALONG L-O-S)	LIS01080
	WRITE(IOOUT,720)	LIS01090
720	FORMAT(1H ,20X,10H-----,3X,14H-----,3X,9H-----,	LIS01100
	+3X,15H-----,3X,13H-----,3X,13H-----)	LIS01110
	IF(Nlayer.EQ.0)GO TO 735	LIS01120
	BLANK=0.0	LIS01130
	DO 730 N=1,Nlayer	LIS01140
	IND=ILCTYP(N)	LIS01150
730	WRITE(IOOUT,760)ALPH(IND),N,ZLBASE(N),ZLTHIC(N),BLANK,TAU(N),T(N)	LIS01160
735	NLO=Nlayer+1	LIS01170
740	IF(NN.EQ.Nlayer)GO TO 800	LIS01180
	LL=0	LIS01190
	DO 750 N=NLO,NN	LIS01200
	LL=LL+1	LIS01210
	IND=ICCTYP(LL)	LIS01220
750	WRITE(IOOUT,760)ALPH(IND),N,ZCBASE(LL),ZCTHIC(LL),RADICL(LL),	LIS01230
	+TAU(N),T(N)	LIS01240
760	FORMAT(1H0,22X,A2,1H/,12,10X,F7.3,7X,F7.3,8X,F7.3,10X,F7.2,7X,	LIS01250
	+E11.5)	LIS01260
	GO TO 800	LIS01270
780	WRITE(IOOUT,790)	LIS01280
790	FORMAT(1H0,20X,59HNO CLOUD OBSCURATION : L-O-S DOES NOT INTERSECT	LIS01290
	+ANY CLOUDS ///	LIS01300
800	RETURN	LIS01310
	END	LIS01320

<pre> SUBROUTINE DEFSET(ISTEP,NMAX) C**** THIS ROUTINE RESETS THE CLOUD BASE HEIGHT, THICKNESS, C**** AND RADIUS VALUES TO THE NEAREST REALISTIC BOUNDARIES C**** IF THEY DO NOT LIE CLOSE TO THE RANGES SPECIFIED IN C**** R. D. M. LOW'S PAPER. C COMMON /BASTOP/ZBAS,ZTOP,ICL,IWV,ILCTYP(10),ICCTYP(10) COMMON /BASTH/ZLBASE(10),ZLTHIC(10),ZCBASE(10),ZCTHIC(10) *,RADICL(10) IF(ISTEP.NE.1)GO TO 10 DO 5 N=1,NMAX ITYPE=ILCTYP(N) IF(ITYPE.EQ.2)GO TO 3 1 IF(ZLBASE(N).LT.0.1)ZLBASE(N)=0.1 IF(ZLBASE(N).GT.1.5)ZLBASE(N)=1.5 IF(ITYPE.EQ.3)GO TO 4 2 IF(ZLTHIC(N).LT.0.2)ZLTHIC(N)=0.2 IF(ZLTHIC(N).GT.1.0)ZLTHIC(N)=1.0 GO TO 5 3 IF(ZLBASE(N).LT.2.0)ZLBASE(N)=2.0 IF(ZLBASE(N).GT.5.0)ZLBASE(N)=5.0 4 IF(ZLTHIC(N).LT.1.0)ZLTHIC(N)=1.0 IF(ZLTHIC(N).GT.4.0)ZLTHIC(N)=4.0 5 CONTINUE 10 IF(ISTEP.NE.2)GO TO 20 DO 15 N=1,NMAX IF(ZCBASE(N).LT.0.8)ZCBASE(N)=0.8 IF(ZCBASE(N).GT.1.5)ZCBASE(N)=1.5 IF(ZCTHIC(N).LT.0.2)ZCTHIC(N)=0.2 IF(ZCTHIC(N).GT.5.0)ZCTHIC(N)=5.0 IF(RADICL(N).LT.0.05)RADICL(N)=0.05 IF(RADICL(N).GT.0.6)RADICL(N)=0.6 15 CONTINUE 20 RETURN END </pre>	<pre> DEF00010 DEF00020 DEF00030 DEF00040 DEF00050 DEF00060 DEF00070 DEF00080 DEF00090 DEF00100 DEF00110 DEF00120 DEF00130 DEF00140 DEF00150 DEF00160 DEF00170 DEF00180 DEF00190 DEF00200 DEF00210 DEF00220 DEF00230 DEF00240 DEF00250 DEF00260 DEF00270 DEF00280 DEF00290 DEF00300 DEF00310 DEF00320 DEF00330 DEF00340 DEF00350 DEF00360 </pre>
--	--

```

SUBROUTINE CLREAD(NLAYER,NCCLDS,IERR)
DIMENSION IALPHA(18),DATELT(5)
DIMENSION X1(20),Y1(20),Z1(20),X0(20),Y0(20),Z0(20)
COMMON /PATHL/X0,Y0,Z0,X1,Y1,Z1,XS,YS,ZS,XT,YT,ZT,ATT(20)
COMMON /GEOMET/PTS(15),IGEOSW
COMMON /BASTOP/ZBAS,ZTOP,ICL,IWV,ILCTYP(10),ICCTYP(10)
COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
COMMON /BASTH/ZLBASE(10),ZLTHIC(10),ZCBASE(10),ZCTHIC(10)
+ ,RADICL(10)
COMMON /INTCL/XCLOUD(10),YCLOUD(10),NLINT(10),NCINT(10)
DATA IALPHA/2HCL,2HST,2HCL,2HAS,2HCL,2HNS,2HCL,2HSC,2HCL,2HCH,
+ 2HCL,2HCC,2HSE,2HEK,2HTA,2HRG,2HGO,2H /
ISFLAG=0
ITFLAG=0
IF(IGEOSW.NE.1)GO TO 60
XS=PTS(4)
YS=PTS(5)
ZS=PTS(6)
XT=PTS(1)
YT=PTS(2)
ZT=PTS(3)
60 CONTINUE
DO 900 N=1,23
READ(IOIN,100)IALPH,IALP,(DATELT(1),I=1,5)
100 FORMAT(2A2,1X,5(E10.5,1X))
INDEX=10
DO 200 K=1,17,2
IF((IALPH.EQ.IALPHA(K)).AND.(IALP.EQ.IALPHA(K+1))) GO TO 50
GO TO 200
50 REALK=FLOAT(K)
ROT=REALK/2.0
KJ=IFIX(ROT)+1
INDEX=KJ
GO TO 300
200 CONTINUE
300 IF(INDEX.LT.7)GO TO (350,350,350,350,400,400),INDEX
INM6=INDEX-6
GO TO (500,600,999,997),INM6
350 NLAYER=NLAYER+1
IF(NLAYER.GT.10) GO TO 993
ILCTYP(NLAYER)=INDEX
ZLBASE(NLAYER)=DATELT(1)
ZLTHIC(NLAYER)=DATELT(2)
GO TO 900
400 NCCLDS=NCCLDS+1
IF(NCCLDS.GT.10) GO TO 995
ICCTYP(NCCLDS)=INDEX
ZCBASE(NCCLDS)=DATELT(1)
ZCTHIC(NCCLDS)=DATELT(2)
RADICL(NCCLDS)=DATELT(3)
XCLOUD(NCCLDS)=DATELT(4)
YCLOUD(NCCLDS)=DATELT(5)
GO TO 900
500 IF(ISFLAG.EQ.1) GO TO 997
XS=DATELT(1)
YS=DATELT(2)
ZS=DATELT(3)
ISFLAG=1
GO TO 900
600 IF(ITFLAG.GT.1) GO TO 997
XT=DATELT(1)
YT=DATELT(2)
ZT=DATELT(3)
ITFLAG=1
900 CONTINUE
GO TO 999
993 IERR=1
WRITE(IOOUT,994)
994 FORMAT(1H0,20X,76H***CLREAD ERROR*** NUMBER OF STRATIFORM CLOUDS I
+ NPUT EXCEEDS THE LIMIT OF 10 //)

```

```

CLR00010
CLR00020
CLR00030
CLR00040
CLR00050
CLR00060
CLR00070
CLR00080
CLR00090
CLR00100
CLR00110
CLR00120
CLR00130
CLR00140
CLR00150
CLR00160
CLR00170
CLR00180
CLR00190
CLR00200
CLR00210
CLR00220
CLR00230
CLR00240
CLR00250
CLR00260
CLR00270
CLR00280
CLR00300
CLR00310
CLR00320
CLR00330
CLR00340
CLR00350
CLR00360
CLR00370
CLR00380
CLR00390
CLR00400
CLR00410
CLR00420
CLR00430
CLR00440
CLR00450
CLR00460
CLR00470
CLR00480
CLR00490
CLR00500
CLR00510
CLR00520
CLR00530
CLR00540
CLR00550
CLR00560
CLR00570
CLR00580
CLR00590
CLR00600
CLR00610
CLR00620
CLR00630
CLR00640
CLR00650
CLR00660

```

GO TO 999	CLR00670
995 IERR=1	CLR00680
WRITE(100UT,996)	CLR00690
996 FORMAT(1H0,20X,76H***CLREAD ERROR*** NUMBER OF CUMULIFORM CLOUDS I	CLR00700
+NPUT EXCEEDS THE LIMIT OF 10 //)	CLR00710
GO TO 999	CLR00720
997 IERR=1	CLR00730
WRITE(100UT,998)	CLR00740
998 FORMAT(1H0,20X,64H***CLREAD ERROR*** IMPROPER INPUT FORMAT OR ABSE	CLR00750
+NT GO SENTINEL //)	CLR00760
999 RETURN	CLR00770
END	CLR00780

```

SUBROUTINE SCREEN(IERR,ICLMAT)
THIS MODULE 1) COMPUTES TRANSMITTANCE REQUIRED TO REDUCE THE
              PROBABILITY OF STATIC TARGET DETECTION BELOW A
              GIVEN LEVEL FOR CERTAIN TYPES OF IMAGERS. (ITAM)
- AND/OR -
              2) COMPUTES HC AND WP 105 MM/155 MM SMOKE MUNITIONS
              REQUIRED TO PRODUCE A SMOKE SCREEN OF USER-DEFINED
              LENGTH AND DURATION FOR VISIBLE, NEAR, MID AND FAR
              IR WAVELENGTHS.

SUBROUTINES CALLED BY SCREEN ARE - CWIC AND ITAM.

ALL OUTPUT FROM SCREEN IS TABULAR.

THE PRESENT VERSION OF CWIC USES THE XSCALE MODULE TO OPTIONALLY
CORRECT FOR EXTINCTION DUE TO FOG, RAIN AND/OR SNOW AT IR
WAVELENGTHS BASED ON VISIBILITY IN THE .5 MICROMETER REGION.
CWIC CAN ALSO COMPUTE PASQUILL CATEGORY FROM FUNDAMENTAL MET INPUTS
IF THE USER CHOOSES NOT TO PROVIDE THE CATEGORY DIRECTLY.
CLIMATOLOGICAL VALUES FROM THE CLIMAT MODULE CAN OPTIONALLY BE
USED AS "TYPICAL" MET INPUTS.

THE ITAM MODULE CAN BE USED IN A LOOPING MODE THROUGH MULTIPLE SETS
OF INPUT RECORDS TO GENERATE TABLES. THE LAST VALUE OF TRANS-
MITTANCE COMPUTED IS THAT VALUE WHICH CAN (OPTIONALLY) BE PASSED
TO CWIC AS THE THRESHOLD LEVEL FOR TOTAL PATH TRANSMITTANCE.

TWO RECORDS MUST BE PROVIDED TO SCREEN:
ONE CARD MUST BE INPUT TO SCREEN TO SELECT OPTIONS: (3(1X,I1))
COL 2  ICITAM  =1 CALL ITAM, OR 0 (NO CALL).
COL 4  ICCWIC  =1 CALL CWIC, OR 0 (NO CALL).
COL 6  ICCLIM  =1 USE CLIMAT FOR MET INPUTS, OR 0 USE USER VALUES.

IF CHOSEN, ALL INPUT RECORDS TO ITAM ARE READ FOLLOWING THE ABOVE
RECORD. (SEE INPUT DESCRIPTION IN ITAM)

AND, IF CHOSEN, INPUT RECORDS TO CWIC ARE THEN READ. (SEE INPUT
DESCRIPTION IN CWIC)

FINALLY, ONE RECORD IS READ BY SCREEN WITH THE WORD "END" IN COLUMNS
1-3. THIS RETURNS CONTROL TO THE EOSAEL EXECUTIVE MODULE.

COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
READ (IOIN,10) ICITAM,ICCWIC,ICCLIM
10  FORMAT(1X,I1,1X,I1,1X,I1)
    IF (ICITAM.NE.0) CALL ITAM(IERR,TFNL)
    IF (IERR.EQ.1) GOTO 15
    IF (ICCLIM.EQ.0) GOTO 20
    IF (ICLMAT.EQ.1) GOTO 20
    IERR=1
    WRITE (IOOUT,30)
30  FORMAT(1X,62H*** IN SCREEN ROUTINE, MET SOURCE SPECIFIED AS CLIMAT
*OLOGICAL,75X,39HBT CLIMAT ROUTINE HAD NOT BEEN CALLED.)
    RETURN
15  WRITE (IOOUT,40)
    TFNL=1
40  FORMAT(1X,21HIERR FLAG SET IN ITAM)
20  IF (ICCWIC.NE.0) CALL CWIC(IERR,ICITAM,ICCLIM,TFNL)
    RETURN
END

```

SCR00010
SCR00020
SCR00030
SCR00040
SCR00050
SCR00060
SCR00070
SCR00080
SCR00090
SCR00100
SCR00110
SCR00120
SCR00130
SCR00140
SCR00150
SCR00160
SCR00170
SCR00180
SCR00190
SCR00200
SCR00210
SCR00220
SCR00230
SCR00240
SCR00250
SCR00260
SCR00270
SCR00280
SCR00290
SCR00300
SCR00310
SCR00320
SCR00330
SCR00340
SCR00350
SCR00360
SCR00370
SCR00380
SCR00390
SCR00400
SCR00410
SCR00420
SCR00430
SCR00440
SCR00450
SCR00460
SCR00470
SCR00480
SCR00490
SCR00500
SCR00510
SCR00520
SCR00530
SCR00540
SCR00550
SCR00560
SCR00570
SCR00580
SCR00590
SCR00600
SCR00610
SCR00620
SCR00630

SUBROUTINE CWIC (IERR, ICITAM, ICCLIM, TFNL)
 CWIC COMPUTES THE REQUIRED SMOKE MUNITIONS EXPENDITURE (NUMBER,
 RATE OF FIRE, PLACEMENT) TO PRODUCE A SCREEN OF DEFINED
 LENGTH AND DURATION USING HC OR WP 105MM OR 155MM SMOKE
 MUNITIONS. WAVELENGTHS ARE FOR RANGES OF VISIBLE, NEAR,
 MID AND FAR IR.

ON INPUT, IF ICITAM IS NON-ZERO, ALL COMPUTATIONS WILL USE
 THE INPUT TFNL AS THE TOTAL SCREEN TRANSMISSION THRESHOLD. IF
 ICITAM IS 0, THEN BUILT-IN TOTAL THRESHOLDS OF .05 ARE USED
 FOR ALL WAVELENGTH REGIONS.

INPUTS TO CWIC ARE ON STANDARDIZED RECORDS CONTAINING KEY-WORDS
 IN COLUMNS 1-4 AND REAL (IE DECIMAL) VALUES FOR ALL INPUTS,
 PLACED IN FIELDS 11-20, 21-30, ..., 71-80. KEY-WORD TYPES
 ARE SCRN FOR SCREEN, OBSERVER/TARGET LOS AND ADVERSE
 WEATHER CORRECTIONS.
 METR FOR METEOROLOGICAL CONDITIONS (MAY BE OMITTED
 IF ICCLIM IS NON-ZERO, IN WHICH CASE THE
 CLIMATOLOGICAL VALUES FROM CLIMAT ARE USED.
 PASQ FOR (OPTIONAL) MET PARAMETERS REQUIRED TO COMPUTE
 THE PASQUILL STABILITY CATEGORY. NOT REQUIRED
 IF PASQUILL CATEGORY ITSELF IS INPUT.
 DONE WHICH RETURNS EXECUTIVE CONTROL BACK TO THE SCREEN
 MODULE.

THE ORDER OF THE INPUT RECORDS IS IMMATERIAL, EXCEPT THAT THE (DONE)
 CARD MUST BE LAST.

INPUTS: (ALL VALUES REAL) FORMAT (2A2,6X,7F10.3)

KEYWORD	COLS.	VARIABLE	DESCRIPTION
SCRN	1-4		SCREEN/LOS DEFINITION (REQUIRED)
	11-20	TIME	- SCREEN DURATION (MINUTES)
	21-30	X0	- SCREEN LENGTH (METERS)
	31-40	H3	- SLANT RANGE OBS-TARGET (KM)
	41-50	AST	- ELEVATION ANGLE OF TARGET FROM OBSERVER WRT HORIZONTAL (DEG.)
	51-60	DLS	- COMPASS DIRECTION (CLOCKWISE WRT NORTH) OF LINE-OF-SIGHT (DEG.)
	61-70	ARE	- TERRAIN ROUGHNESS ELEMENT (CM)
	71-80	FOG	- ADVERSE WEATHER/HAZE SELECTION 0. = NO ADVERSE WEATHER 1. = ONLY CORRECT VISIBLE WAVE- LENGTHS FOR INPUT VISIBILITY. 2. = CORRECT FOR FOG/HAZE FOR MARITIME ARCTIC AIR MASS 3. = CORRECT FOR FOG/HAZE FOR MARITIME POLAR AIR MASS 4. = CORRECT FOR FOG/HAZE FOR CONTINENTAL POLAR AIR MASS 5. = CORRECT FOR RAIN. 6. = CORRECT FOR SNOW.
METR	1-4		MET INPUTS (NOT REQUIRED IF ICCLIM NON-ZERO FROM SCREEN MODULE.)
	11-20	S3	- WINDSPEED (METERS/SEC)
	21-30	D0	- WIND DIRECTION (DEG) CLOCKWISE WRT NORTH, USUAL MET CONVENTION
	31-40	PCAT	- PASQUILL CATEGORY. IF INPUT AS 0, SEE PASQ RECORD BELOW. OTHERWISE, 1.=A, 2.=B, 3.=C, 4.=D, 5.=E, 6.=F
	41-50	VS	- VISIBILITY (KM) NOT REQUIRED IF FOG = 0. IS SPECIFIED.
	51-60	R0	- RELATIVE HUMIDITY (PERCENT). IF 0, THEN DEW POINT AND TEMPERATURE ARE REQUIRED BELOW TO COMPUTE R0.
	61-70	T0	- AIR TEMP. (DEG C) REQUIRED IF R0 NOT GIVEN.

900	FORMAT(2A2,6X,7F10.3)	CWC01400
901	FORMAT(1X,51HIN CWIC, THE FOLLOWING CARD DOES NOT CONFORM TO PRO,	CWC01410
	*15HPER CONVENTIONS/1X,2A2,6X,7F10.3)	CWC01420
	IF (ICOU.LE.4) GOTO 10	CWC01430
	WRITE (ICOUT,902)	CWC01440
902	FORMAT(1X,39HINVALID INPUTS TO CWIC. IERR=1 RETURNED)	CWC01450
	IERR=1	CWC01460
	RETURN	CWC01470
20	TIME=RV(1)	CWC01480
	X0=RV(2)	CWC01490
	H3=RV(3)	CWC01500
	AST=RV(4)	CWC01510
	DLS=RV(5)	CWC01520
	ARE=RV(6)	CWC01530
	FOG=RV(7)	CWC01540
	IF (IGEOSW.NE.1) GO TO 22	CWC01550
	DELX=PTS(1)-PTS(4)	CWC01560
	DELY=PTS(2)-PTS(5)	CWC01570
	DELZ=PTS(3)-PTS(6)	CWC01580
	H3=SQRT(DELX**2+DELY**2+DELZ**2)	CWC01590
	HDIS=SQRT(DELX**2+DELY**2)	CWC01600
	RTDCON=57.29577951	CWC01610
	AST=RTDCON*ACOS(HDIS/H3)	CWC01620
	IF (HDIS.GT.1.E-20) DLS=RTDCON*ACOS(DELY/HDIS)	CWC01630
	IF (DELX.LT.0.) DLS=360.-DLS	CWC01640
22	CONTINUE	CWC01650
	GOTO 10	CWC01660
40	S3=RV(1)	CWC01670
	D0=RV(2)	CWC01680
	PCAT=RV(3)	CWC01690
	VS=RV(4)	CWC01700
	R0=RV(5)	CWC01710
	T0=RV(6)	CWC01720
	T1=RV(7)	CWC01730
	GOTO 10	CWC01740
50	SLAT=RV(1)	CWC01750
	SLONG=RV(2)	CWC01760
	SJDATE=RV(3)	CWC01770
	SZHOUR=RV(4)	CWC01780
	C0=RV(5)	CWC01790
	C1=RV(6)	CWC01800
	GOTO 10	CWC01810
C***	BEGIN COMPUTATIONS. -- FIRST MET VALUES.	CWC01820
60	IF (ICCLIM.EQ.0) GOTO 70	CWC01830
C***	USE CLIMAT PASSED VALUES:	CWC01840
	T0=TEMP	CWC01850
	T1=DP	CWC01860
	R0=RH	CWC01870
	PCAT=FLOAT(IPASCT)	CWC01880
	VS=VIS	CWC01890
	D0=WNDDIR	CWC01900
	S3=WNDEL	CWC01910
C***	PROVIDE WINDSPEED IN KNOTS AS S0:	CWC01920
70	S0=S3/.515	CWC01930
	IF (S0.LE.1.) S3=.515	CWC01940
	IF (S0.LE.1.) S0=1.	CWC01950
C***	NOW CHECK RH AND COMPUTE IF NECESSARY.	CWC01960
	IRNOT=0	CWC01970
	IF (R0.GT.0.) GOTO 80	CWC01980
	IF (T0.GT.0.) GO TO 76	CWC01990
	A0=9.5	CWC02000
	B0=265.5	CWC02010
	IF (T0.LE.0.) GO TO 78	CWC02020
76	CONTINUE	CWC02030
	A0=7.5	CWC02040
	B0=237.3	CWC02050
78	CONTINUE	CWC02060
	IF (T1.GE.0.) GO TO 79	CWC02070
	A1=9.5	CWC02080
	B1=265.5	CWC02090

79	IF(T1.LE.0.) GO TO 75	CWC02100
	CONTINUE	CWC02110
	A1=7.5	CWC02120
	B1=237.3	CWC02130
75	CONTINUE	CWC02140
	E0=6.11*10.**((A0*T0)/(B0+T0))	CWC02150
	E1=6.11*10.**((A1*T1)/(B1+T1))	CWC02160
	R0=(E1/E0)*100.	CWC02170
	IRNOT=1	CWC02180
C***	NOW CHECK PASQUILL CATEGORY..	CWC02190
80	IP0=IFIX(PCAT+.001)	CWC02200
	IF (IP0.GT.6) IP0=6	CWC02210
	IPNOT=0	CWC02220
	IF (IP0.GT.0) GOTO 90	CWC02230
	CALL CWIC1 (IP0,C0,C1,SLAT,SLONG,SJDATE,SZHOURL,S0)	CWC02240
	IPNOT=1	CWC02250
C***	NEXT COMPUTE CL VALUE FOR THRESHOLD TRANSMITTANCE, CORRECTED FOR	CWC02260
C	FOG,RAIN,...	CWC02270
C***	VISIBILITY EXTINCTION (KM-1) .. NOTE THAT IF VS=0., THEN SET TO	CWC02280
C	CLEAR DAY AND COMPUTATIONS CONTINUE.	CWC02290
90	EX55=0.	CWC02300
	IF (VS.GT.0.) EX55=3.912/VS	CWC02310
C***	TRANSMISSION THRESHOLDS..	CWC02320
	DO 92 I=1,4	CWC02330
	T(I)=TRSH(I)	CWC02340
	IF (ICITAM.NE.0) T(I)=TFNL	CWC02350
	TR(I)=T(I)	CWC02360
92	CONTINUE	CWC02370
C***	CORRECTIONS FOR WEATHER	CWC02380
	IFOG=-1+IFIX(FOG+.0001)	CWC02390
	IF (IFOG.LE.-1) GOTO 100	CWC02400
C***	CORRECT VISIBLE FOR VISIBILITY (H3 SLANT RNG, AST ELEV. ANG.)	CWC02410
	XSTRN=0.	CWC02420
	EPTH=EX55*H3	CWC02430
	IF (EPTH.LT.12.) XSTRN=EXP(-EPTH)	CWC02440
C SET	VSET FOR NO CARD I/O IN XSCALE	CWC02450
	VSET=89.	CWC02460
	IF (AST.GT.0.1 .OR. AST.LT.-0.1) AND. IFOG.LT.5)	CWC02470
	*CALL XSCALE(WAVE(I),VSET,EX55,XSTRN,IERR,ISLANT(I),IFOG,H3,AST)	CWC02480
	IF (XSTRN.LE.0.) XSTRN=.0001	CWC02500
	IF (XSTRN.GT.1.) XSTRN=1.	CWC02510
	T(I)=T(I)/XSTRN	CWC02520
	IF (IFOG.EQ.0) GOTO 100	CWC02530
C	CORRECT NON-VISIBLE FOR SEEABILITY.	CWC02540
94	CONTINUE	CWC02550
	DO 96 I=2,4	CWC02560
	ISLNT=ISLANT(I)	CWC02570
	IF (AST.EQ.0.) ISLNT=0	CWC02580
	IF (IFOG.GE.5) ISLNT=0	
	CALL XSCALE(WAVE(I),VSET,EX55,XSTRN,IERR,ISLNT,IFOG,H3,AST)	CWC02590
	IF (XSTRN.LE.0.) XSTRN=.0001	CWC02610
	IF (XSTRN.GT.1.) XSTRN=1.	CWC02620
	T(I)=T(I)/XSTRN	CWC02630
96	CONTINUE	CWC02640
C***	COMPUTE CL FOR THRESHOLD TRANSMISSION REQUIRED OF SMOKE.	CWC02650
100	DO 108 I=1,4	CWC02660
	IF (T(I).GT.1.) T(I)=1.	CWC02670
	IF (T(I).LE.0.) T(I)=.00001	CWC02680
	DO 105 J=1,2	CWC02690
	C(I,J)=ALOG(T(I))/(-CS(I,J))	CWC02700
105	CONTINUE	CWC02710
108	CONTINUE	CWC02720
C***	ATMOSPHERIC DIFFUSION	CWC02730
	CALL CWIC3 (ARE,DLS,C2,D0,D2,H,IP0,R0,R2,U,V,Y1)	CWC02740
C***	MUNITIONS EXPENDITURES	CWC02750
	CALL CWIC4 (C,C2,D2,H,P,Q1,Q2,R,R2,S3,TIME,V,X0,Y,Y1,Z,IP0)	CWC02760
C**	END MAIN CWIC COMPUTATIONS, FINAL OUTPUT.	CWC02770
	IF (ICITAM.GT.0) WRITE(10000)	CWC02780
	WRITE(10000,10200)	CWC02790
	WRITE(10000,10800)	CWC02800

WRITE (IOOUT,10140)	CWC02810
WRITE (IOOUT,10300)	CWC02820
WRITE (IOOUT,10400)	CWC02830
WRITE (IOOUT,10200)	CWC02840
WRITE (IOOUT,14000) H3	CWC02850
WRITE (IOOUT,14100) AST	CWC02860
WRITE (IOOUT,14200) DLS	CWC02870
WRITE (IOOUT,11900) ARE	CWC02880
WRITE (IOOUT,11400)	CWC02890
IAD=1	CWC02900
IF (IFOG.GE.0) IAD=3	CWC02910
WRITE (IOOUT,14300) JANS(IAD), JANS(IAD+1)	CWC02920
IAD=1	CWC02930
IF (IFOG.EQ.1) IAD=3	CWC02940
WRITE (IOOUT,14400) JANS(IAD), JANS(IAD+1)	CWC02950
IAD=1	CWC02960
IF (IFOG.EQ.2) IAD=3	CWC02970
WRITE (IOOUT,14500) JANS(IAD), JANS(IAD+1)	CWC02980
IAD=1	CWC02990
IF (IFOG.EQ.3) IAD=3	CWC03000
WRITE (IOOUT,14600) JANS(IAD), JANS(IAD+1)	CWC03010
IAD=1	CWC03020
IF (IFOG.EQ.4) IAD=3	CWC03030
WRITE (IOOUT,14700) JANS(IAD), JANS(IAD+1)	CWC03040
IAD=1	CWC03050
IF (IFOG.EQ.5) IAD=3	CWC03060
WRITE (IOOUT,14800) JANS(IAD), JANS(IAD+1)	CWC03070
WRITE (IOOUT,15000)	CWC03080
WRITE (IOOUT,11800) S3	CWC03090
WRITE (IOOUT,11700) D0	CWC03100
JP=JPASCT(IP0)	CWC03110
WRITE (IOOUT,12000) JP	CWC03120
WRITE (IOOUT,11300) VS	CWC03130
WRITE (IOOUT,12100) R0	CWC03140
IF (IRNOT.EQ.1) WRITE (IOOUT,11500) T0	CWC03150
IF (IRNOT.EQ.1) WRITE (IOOUT,11600) T1	CWC03160
IF (IPNOT.EQ.0) GOTO 120	CWC03170
WRITE (IOOUT,15100)	CWC03180
IF (SLAT.GE.0.) WRITE (IOOUT,10601) SLAT	CWC03190
SLATI=-SLAT	CWC03200
IF (SLAT.LT.0.) WRITE (IOOUT,10602) SLATI	CWC03210
IF (SLONG.GE.0.) WRITE (IOOUT,10701) SLONG	CWC03220
SLONGI=-SLONG	CWC03230
IF (SLONG.LT.0.) WRITE (IOOUT,10702) SLONGI	CWC03240
WRITE (IOOUT,10900) SJDATE	CWC03250
WRITE (IOOUT,11000) SZHOUR	CWC03260
WRITE (IOOUT,11100) C0	CWC03270
WRITE (IOOUT,11200) C1	CWC03280
120 CONTINUE	CWC03290
WRITE (IOOUT,15200)	CWC03300
DO 150 I=1,4	CWC03310
IWL=4*(I-1)	CWC03320
WRITE (IOOUT,15300) (LNGTH(IWL+J), J=1,4), TR(I), T(I)	CWC03330
150 CONTINUE	CWC03340
WRITE (IOOUT,10000)	CWC03350
C* PRINT MUNITION EXPENDITURES	CWC03360
WRITE (IOOUT,10130)	CWC03370
WRITE (IOOUT,16200) (LNGTH(J), J=1,8)	CWC03380
WRITE (IOOUT,10100)	CWC03390
WRITE (IOOUT,16300)	CWC03400
WRITE (IOOUT,16400) X0, TIME, X0, TIME	CWC03410
WRITE (IOOUT,10200)	CWC03420
WRITE (IOOUT,16500) ISMOKE(1), ISMOKE(1)	CWC03430
WRITE (IOOUT,10100)	CWC03440
WRITE (IOOUT,16600) IGUN(1), IGUN(2), IGUN(1), IGUN(2)	CWC03450
WRITE (IOOUT,10100)	CWC03460
WRITE (IOOUT,16700)	CWC03470
WRITE (IOOUT,16800)	CWC03480
WRITE (IOOUT,16900) Q1(1,1,1), Y(1,1,1), Q1(2,1,1), Y(2,1,1)	CWC03490
WRITE (IOOUT,17000) Q1(1,2,1), R(1,1), Y(1,2,1), P(1,1,1), Q1(2,2,1),	CWC03500

```

*R(2,1),Y(2,2,1),P(2,1,1)
WRITE(1000,10200)
WRITE(1000,16600) IGUN(3),IGUN(4),IGUN(3),IGUN(4)
WRITE(1000,10100)
WRITE(1000,16700)
WRITE(1000,16800)
WRITE(1000,16900) Q1(1,1,2),Y(1,1,2),Q1(2,1,2),Y(2,1,2)
WRITE(1000,17000) Q1(1,2,2),R(1,1),Y(1,2,2),P(1,2,1),Q1(2,2,2),
*R(2,1),Y(2,2,2),P(2,2,1)
WRITE(1000,10200)
WRITE(1000,16500) ISMOKE(2),ISMOKE(2)
WRITE(1000,10100)
WRITE(1000,16600) IGUN(1),IGUN(2),IGUN(1),IGUN(2)
WRITE(1000,10100)
WRITE(1000,16700)
WRITE(1000,16800)
WRITE(1000,16900) Q2(1,1,1),Z(1,1,1),Q2(2,1,1),Z(2,1,1)
IF(1P0.GT.4) R(1,2)=2.
IF(1P0.GT.4) R(2,2)=2.
WRITE(1000,17000) Q2(1,2,1),R(1,2),Z(1,1,1),P(1,1,2),Q2(2,2,1),
*R(2,2),Z(2,1,1),P(2,1,2)
WRITE(1000,10200)
WRITE(1000,16600) IGUN(3),IGUN(4),IGUN(3),IGUN(4)
WRITE(1000,10100)
WRITE(1000,16700)
WRITE(1000,16800)
WRITE(1000,16900) Q2(1,1,2),Z(1,2,2),Q2(2,1,2),Z(2,2,2)
IF(1P0.GT.4) R(1,2)=1.
IF(1P0.GT.4) R(2,2)=1.
WRITE(1000,17000) Q2(1,2,2),R(1,2),Z(1,2,2),P(1,2,2),Q2(1,2,2),
*R(2,2),Z(2,2,2),P(2,2,2)
WRITE(1000,10000)
WRITE(1000,10130)
DO 7100 I=3,4
IWL=4*(I-1)
J1=IWL+1
J2=IWL+4
WRITE(1000,12200) (LNGTH(J),J=J1,J2)
WRITE(1000,10100)
WRITE(1000,12300)
WRITE(1000,12400) X0,TIME
WRITE(1000,10200)
WRITE(1000,12500) ISMOKE(2)
WRITE(1000,10100)
WRITE(1000,13100)
WRITE(1000,13200)
WRITE(1000,13300) IGUN(1),IGUN(2),Q2(1,2,1),R(1,2),P(1,1,2)
WRITE(1000,13300) IGUN(3),IGUN(4),Q2(1,2,2),R(1,2),P(1,2,2)
IF(1.EQ.3) WRITE(1000,10140)
IF(1.EQ.4) WRITE(1000,10000)
7100 CONTINUE
RETURN
C* FORMAT STATEMENTS.
10000 FORMAT(1H1)
10100 FORMAT(1H )
10200 FORMAT(1H0)
10130 FORMAT(///)
10140 FORMAT(////)
10300 FORMAT(55X,21HMUNITION EXPENDITURES)
10400 FORMAT(56X,19HFOR HC AND WP SMOKE)
10601 FORMAT(45X,35HLATITUDE - DEG =,F7.2,6H NORTH)
10602 FORMAT(45X,35HLATITUDE - DEG =,F7.2,6H SOUTH)
10701 FORMAT(45X,35HLONGITUDE - DEG =,F7.2,6H EAST)
10702 FORMAT(45X,35HLONGITUDE - DEG =,F7.2,6H WEST)
10800 FORMAT(1H0,47X,36(1H*)/48X,1H*,34X,1H*/48X,1H*,4X,
*26HCWIC MUNITION EXPENDITURES,4X,1H*/48X,1H*,34X,1H*/
*48X,36(1H*))
10900 FORMAT(45X,35HJULIAN DATE - DAY =,F7.0)
11000 FORMAT(45X,35HGMT TIME - HOUR =,F7.2)
11100 FORMAT(45X,35HCEILING - METERS =,F7.1)

```

11200	FORMAT(45X,35HCLOUD COVER	- PERCENT	=,F7.1)	CWC04210
11300	FORMAT(45X,35HVISIBILITY	- KM	=,F7.3)	CWC04220
11400	FORMAT(1H0,44X,35HATMOSPHERIC EXTINCTION CORRECTIONS)			CWC04230
11500	FORMAT(45X,35HTEMPERATURE	- DEG C	=,F7.1)	CWC04240
11600	FORMAT(45X,35HDEW POINT	- DEG C	=,F7.1)	CWC04250
11700	FORMAT(45X,35HWIND DIRECTION	- DEG	=,F7.2)	CWC04260
11800	FORMAT(45X,35HWINDSPEED	- M/SEC	=,F7.2)	CWC04270
11900	FORMAT(45X,35HAVG ROUGHNESS ELEMENT	- CM	=,F7.1)	CWC04280
12000	FORMAT(45X,35HPASQUILL CATEGORY	-	=,5X,A2)	CWC04290
12100	FORMAT(45X,35HRELATIVE HUMIDITY	- PERCENT	=,F7.1)	CWC04300
12200	FORMAT(60X,11(1H-)/60X,1H-,1X,4A2,1H-/60X,11(1H-))			CWC04310
12300	FORMAT(62X,6HLENGTH,8X,8HDURATION/62X,6HMETERS,9X,7HMINUTES)			CWC04320
12400	FORMAT(47X,6HSCREEN,8X,F7.0,9X,F7.2)			CWC04330
12500	FORMAT(59X,A2,13H SMOKE SCREEN/59X,15(1H-))			CWC04340
12600	FORMAT(59X,A2,A1,11HMM HOWITZER)			CWC04350
12700	FORMAT(47X,37HVOLLEY	GUNS RATE	SPACING ROUNDS)	CWC04360
12800	FORMAT(47X,30H	/MIN	METERS)	CWC04370
12900	FORMAT(47X,11HINITIAL:	F5.0,6X,F8.0)		CWC04380
13000	FORMAT(47X,11HSUSTAINING:	F5.0,F5.1,F9.0,F7.0)		CWC04390
13100	FORMAT(47X,31H	ROUNDS/	RATE/ TOTAL)	CWC04400
13200	FORMAT(47X,32H	60 METERS	MINUTE ROUNDS)	CWC04410
13300	FORMAT(47X,A2,A1,5HMM:	F5.0,5X,F4.0,3X,F7.0)		CWC04420
14000	FORMAT(45X,35HSLANT RANGE OBS-TGT	- KM	=,F7.3)	CWC04430
14100	FORMAT(45X,35HELEVATION OF TARGET	- DEG	=,F7.2)	CWC04440
14200	FORMAT(45X,35HAZIMUTH OF TARGET	- DEG	=,F7.2)	CWC04450
14300	FORMAT(45X,35HCORRECTED FOR VISIBILITY	-	,4X,2A2)	CWC04460
14400	FORMAT(45X,35HMARITIME ARCTIC AIR MASS	-	,4X,2A2)	CWC04470
14500	FORMAT(45X,35HMARITIME POLAR AIR MASS	-	,4X,2A2)	CWC04480
14600	FORMAT(45X,35HCONTINENTAL POLAR AIR MASS	-	,4X,2A2)	CWC04490
14700	FORMAT(45X,35HCORRECTED FOR RAIN	-	,4X,2A2)	CWC04500
14800	FORMAT(45X,35HCORRECTED FOR SNOW	-	,4X,2A2)	CWC04510
15000	FORMAT(1H0,44X,35HMETEOROLOGICAL INPUTS)	CWC04520
15100	FORMAT(1H0,44X,35HINPUTS FOR PASQUILL CATEGORY)	CWC04530
15200	FORMAT(1H0,44X,42HTRANSMISSION THRESHOLDS	TOTAL	SMOKE)	CWC04540
15300	FORMAT(45X,4A2,18X,F5.3,6X,F5.3)			CWC04550
16200	FORMAT(33X,12(1H-),44X,12(1H-)/33X,1H-,1X,4A2,1X,1H-,44X,1H-,1X,			CWC04560
	*4A2,1X,1H-/33X,12(1H-),44X,12(1H-))			CWC04570
16300	FORMAT(1X,2(34X,6HLENGTH,8X,8HDURATION)/1X,2(34X,6HMETERS,9X,			CWC04580
	*7HMINUTES))			CWC04590
16400	FORMAT(1X,2(19X,6HSCREEN,8X,F7.0,9X,F7.2))			CWC04600
16500	FORMAT(32X,A2,13H SMOKE SCREEN,41X,A2,13H SMOKE SCREEN/32X,			CWC04610
	*15(1H-),41X,15(1H-))			CWC04620
16600	FORMAT(32X,A2,A1,11HMM HOWITZER,42X,A2,A1,11HMM HOWITZER)			CWC04630
16700	FORMAT(1X,2(19X,6HVOLLEY,6X,25HGUNS RATE	SPACING ROUNDS))		CWC04640
16800	FORMAT(37X,4H/MIN,3X,6HMETERS,43X,4H/MIN,3X,6HMETERS)			CWC04650
16900	FORMAT(20X,11HINITIAL:	F5.0,6X,F8.0,26X,11HINITIAL:		CWC04660
	*F5.0,6X,F8.0)			CWC04670
17000	FORMAT(1X,2(19X,11HSUSTAINING:	F5.0,F5.1,F9.0,F7.0))		CWC04680
	END			CWC04690

	SUBROUTINE CWIC1 (IP0,C0,C1,SLAT,SLONG,SJDATE,SHOUR,S0)	CWX00010
	DIMENSION ITAB(7,9)	CWX00020
	COMMON /CONST/ PI,PI2,PIRAD,TWOPI,TORRMB,CDECK	CWX00030
	DATA ITAB /	CWX00040
1	1,1,2,3,4,6,6,	CWX00050
2	1,2,2,3,4,6,6,	CWX00060
3	1,2,3,4,4,5,6,	CWX00070
4	2,2,3,4,4,5,6,	CWX00080
5	2,2,3,4,4,4,5,	CWX00090
6	2,3,3,4,4,4,5,	CWX00100
7	3,3,4,4,4,4,5,	CWX00110
8	3,3,4,4,4,4,4,	CWX00120
9	3,4,4,4,4,4,4,	CWX00130
	ASIN(ARG)=ATAN2(ARG,SQRT(1.-ARG**2))	
	C*METEOROLOGICAL CALCULATIONS.	CWX00140
C		CWX00150
	IF(C1 .NE. 100.) GO TO 1000	CWX00160
	IF(C0 .GT. 2133.6042) GO TO 1000	CWX00170
	I1=0	CWX00180
	I2=0	CWX00190
	GO TO 2300	CWX00200
1000	CONTINUE	CWX00210
C	CALCULATE ANGULAR FRACTION OF A YEAR FOR A GIVEN JULIAN DATE.	CWX00220
	R9=PIRAD	CWX00230
	D9=1./R9	CWX00240
	SLAT1=SLAT*R9	CWX00250
	A0=((SJDATE-1.)*360.)/365.242	CWX00260
C	CALCULATE SOLAR DECLINATION ANGLE (A4).	CWX00270
	A1=A0*R9	CWX00280
	A2=279.9348+A0	CWX00290
	A2=A2+(1.914827*SIN(A1))-(0.079525*COS(A1))	CWX00300
	A2=A2+(0.019938*SIN(2*A1))-(0.00162*COS(2*A1))	CWX00310
	A2=A2*R9	CWX00320
	A3=23.4438*R9	CWX00330
	A4=SIN(A3)*SIN(A2)	CWX00340
	A4=ASIN(A4)	CWX00350
C	CALCULATE THE TIME OF MERIDIAN PASSAGE - TRUE SOLAR NOON (A5).	CWX00360
	A5= 12.+(0.12357*SIN(A1))-(0.004289*COS(A1))	CWX00370
	A5=A5+(0.153809*SIN(2*A1))+(0.060783*COS(2*A1))	CWX00380
C	CALCULATE SOLAR HOUR ANGLE (A6)*** NOTE THIS VERSION USES + SIGN	CWX00390
C	ON SLONG DUE TO EAST-LONGITUDE POSITIVE CONVENTION.	CWX00400
	A6=15.*(SHOUR-A5)+SLONG	CWX00410
	A6=A6*R9	CWX00420
C	CALCULATE SOLAR ALTITUDE (A7).	CWX00430
	A7=SIN(SLAT1)*SIN(A4)+COS(SLAT1)*COS(A4)*COS(A6)	CWX00440
	A7=ASIN(A7)	CWX00450
1100	CONTINUE	CWX00460
	A7=A7*D9	CWX00470
C	CALCULATE INSOLATION CLASS NUMBER.	CWX00480
	I2=0	CWX00490
	IF(A7 .LE. 60.) GO TO 1200	CWX00500
	I2=4	CWX00510
	GO TO 1500	CWX00520
1200	CONTINUE	CWX00530
	IF(A7 .LE. 35.) GO TO 1300	CWX00540
	I2=3	CWX00550
	GO TO 1500	CWX00560
1300	CONTINUE	CWX00570
	IF(A7 .LE. 15.) GO TO 1400	CWX00580
	I2=2	CWX00590
	GO TO 1500	CWX00600
1400	CONTINUE	CWX00610
	IF(A7 .LE. 0.) GO TO 2200	CWX00620
	I2=1	CWX00630
C	CALCULATE NET RADIATION INDEX FOR DAYTIME.	CWX00640
1500	CONTINUE	CWX00650
	I3=0	CWX00660
	IF(C1 .GT. 50.) GO TO 1600	CWX00670
	I3=I2	CWX00680
	GO TO 1900	CWX00690

```

1600 CONTINUE
    IF(C0 .GT. 2133.6042) GO TO 1700
    I3=I2-2
    GO TO 1900
1700 CONTINUE
    IF(C0 .GE. 4876.8096) GO TO 1800
    I3=I2-1
    GO TO 1900
1800 CONTINUE
    IF(C1 .NE. 100.) GO TO 1900
    I3=I2-1
    GO TO 1900
1900 CONTINUE
    IF(I3 .NE. 0) GO TO 2000
    I3=I2
    GO TO 1900
2000 CONTINUE
    IF(I3 .GT. 1) GO TO 2100
    I3=1
    GO TO 1900
2100 CONTINUE
    I1=I3
    GO TO 2300
C COMPUTE NET RADIATION INDEX FOR NIGHTTIME.
2200 CONTINUE
    IF(C1 .GT. 40.) GO TO 2250
    I1=-2
    GO TO 2300
2250 CONTINUE
    I1=-1
    GO TO 2300
C CALCULATE PASQUILL STABILITY CATAGORY.
2300 CONTINUE
    I4=0
    IF(I1 .NE. 4) GO TO 2400
    I4=1
    GO TO 2400
2400 CONTINUE
    IF(I1 .NE. 3) GO TO 2420
    I4=2
    GO TO 2440
2420 CONTINUE
    IF(I1 .NE. 2) GO TO 2440
    I4=3
    GO TO 2460
2440 CONTINUE
    IF(I1 .NE. 1) GO TO 2460
    I4=4
    GO TO 2480
2460 CONTINUE
    IF(I1 .NE. 0) GO TO 2480
    I4=5
    GO TO 2480
2480 CONTINUE
    IF(I1 .NE. -1) GO TO 2500
    I4=6
    GO TO 2500
2500 CONTINUE
    IF(I1 .NE. -2) GO TO 2520
    I4=7
    GO TO 2520
2520 CONTINUE
    IF(S0 .GE. 2.) GO TO 2540
    I5=1
    GO TO 2700
2540 CONTINUE
    IF(S0 .GE. 4.) GO TO 2560
    I5=2
    GO TO 2700
2560 CONTINUE
    IF(S0 .GE. 6.) GO TO 2580
    I5=3
    GO TO 2700
2580 CONTINUE
    IF(S0 .GE. 7.) GO TO 2600
    I5=4
    GO TO 2700
2600 CONTINUE
    IF(S0 .GE. 8.) GO TO 2620
    I5=5

```

```

CWX00700
CWX00710
CWX00720
CWX00730
CWX00740
CWX00750
CWX00760
CWX00770
CWX00780
CWX00790
CWX00800
CWX00810
CWX00820
CWX00830
CWX00840
CWX00850
CWX00860
CWX00870
CWX00880
CWX00890
CWX00900
CWX00910
CWX00920
CWX00930
CWX00940
CWX00950
CWX00960
CWX00970
CWX00980
CWX00990
CWX01000
CWX01010
CWX01020
CWX01030
CWX01040
CWX01050
CWX01060
CWX01070
CWX01080
CWX01090
CWX01100
CWX01110
CWX01120
CWX01130
CWX01140
CWX01150
CWX01160
CWX01170
CWX01180
CWX01190
CWX01200
CWX01210
CWX01220
CWX01230
CWX01240
CWX01250
CWX01260
CWX01270
CWX01280
CWX01290
CWX01300
CWX01310
CWX01320
CWX01330
CWX01340
CWX01350
CWX01360
CWX01370
CWX01380
CWX01390

```

```

2620 GO TO 2700
      CONTINUE
      IF(S0 .GE. 10.) GO TO 2640
      IS=6
      GO TO 2700
2640 CONTINUE
      IF(S0 .GE. 11.) GO TO 2660
      IS=7
      GO TO 2700
2660 CONTINUE
      IF(S0 .GE. 12.) GO TO 2680
      IS=8
      GO TO 2700
2680 CONTINUE
      IS=9
2700 CONTINUE
      IP0=ITAB(I4,IS)
      RETURN
      END

```

```

CWX01400
CWX01410
CWX01420
CWX01430
CWX01440
CWX01450
CWX01460
CWX01470
CWX01480
CWX01490
CWX01500
CWX01510
CWX01520
CWX01530
CWX01540
CWX01550
CWX01560
CWX01570
CWX01580

```

```

SUBROUTINE CWIC3 (ARE,DLS,C2,D0,D2,H,IP0,R0,R2,U V,Y1)
DIMENSION A(6),S(6,3),D(6,3),H(2,2),U(2,2),V(2)
DIMENSION W(6)
COMMON /CONST/ PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK
DATA A/0.4,0.32,0.22,0.144,0.102,0.076/
DATA S /
*.139085297,.122097643,.110104377,.097649832,.070772166,.055487093,
*.015017284,.010970370,.010962963,.010418519,7.27284E-3,6.55309E-3,
*-1.02581E-4,-6.80135E-5,-6.73401E-5,-6.83502E-5,-4.50056E-5,
*-4.01796E-5/
DATA D /
*.944814815,.894803591,.854792368,.816026936,.786026936,.726015713,
*-4.85185E-3,-4.83951E-3,-4.82716E-3,-6.07407E-3,-6.07407E-3,
*-6.06173E-3
*.3.7037E-5,3.59147E-5,3.47924E-5,4.7138E-5,4.7138E-5,4.60157E-5/
DATA W/0.016,0.016,0.016,0.016,0.016,0.016/
C*ATMOSPHERIC DIFFUSION CALCULATIONS.
A1=-1.24+1.19*ALOG10(ARE)
Z1=10.**A1
A2=ABS(DLS-D0)*(PI/180.)
R2=SQRT(13.69/(13.69*SIN(A2)*SIN(A2)+COS(A2)*COS(A2)))
Y1=1.09521547+(0.02906894*R0)-(4.9575E-04*R0*R0)+
2 (4.82E-06*R0*R0*R0)
2 Y2=3.364059144+(0.060502571*R0)-(1.15301E-03*R0*R0)+
2 (1.33942E-05*R0*R0*R0)
C2=S(IP0,1)+S(IP0,2)*Z1+S(IP0,3)*Z1**2
D1=D(IP0,1)+D(IP0,2)*Z1+D(IP0,3)*Z1**2
D2=1/D1
DO 5400 I=1,4
C*CALCULATE CROSSWIND INTEGRATED CONCENTRATION FOR WP SMOKE.
DO 5300 K=1,2
IF (I.LT. 3 .AND. IP0.GT. 4) GOTO 5300
S1=U(K,1)+0.74*A(IP0)*100.**0.9
S2=U(K,2)+0.667*C2*100.**D1
V(K)=(W(IP0)*Y2*H(K,2))/(PI*S1*S2)
5300 CONTINUE
5400 CONTINUE
RETURN
END

```

```

CWY00010
CWY00020
CWY00030
CWY00040
CWY00050
CWY00060
CWY00070
CWY00080
CWY00090
CWY00100
CWY00110
CWY00120
CWY00130
CWY00140
CWY00150
CWY00160
CWY00170
CWY00180
CWY00190
CWY00200
CWY00210
CWY00220
CWY00230
CWY00240
CWY00250
CWY00260
CWY00270
CWY00280
CWY00290
CWY00300
CWY00310
CWY00320
CWY00330
CWY00340
CWY00350
CWY00360
CWY00370
CWY00380
CWY00390

```

```

SUBROUTINE CWIC4 (C,C2,D2,H,P,Q1,Q2,R,R2,S3,TIME,V,X0,Y,Y1,Z,IP0) CWZ00010
REAL ME(2) CWZ00020
DIMENSION C(4,2),H(2,2),P(4,2,2),Q1(4,2,2),Q2(4,2,2),R(4,2),V(2), CWZ00030
*Y(4,2,2),Z(4,2,2) CWZ00040
DATA ME/0.4,0.4/ CWZ00050
C*MUNITIONS EXPENDITURES. CWZ00060
DO 6900 I=1,4 CWZ00070
C* CALCULATE INITIAL SHELL SPACING FOR HC SMOKE CWZ00080
DO 6800 K=1,2 CWZ00090
IF(I.GT.2) GO TO 6100 CWZ00100
Y(I,1,K)=45.*S3 CWZ00110
C* CALCULATE SUSTAINING SHELL SPACING FOR HC SMOKE CWZ00120
IF(C(I,1).NE.0.) GO TO 5500 CWZ00130
Y(I,2,K)=0. CWZ00140
GO TO 5600 CWZ00150
5500 CONTINUE CWZ00160
Y(I,2,K)=1/R2*((0.731*ME(K)*Y1*H(K,1))/(C2*S3*C(I,1)))*D2 CWZ00170
IF(Y(I,2,K).GT.X0) Y(I,2,K)=X0 CWZ00180
5600 CONTINUE CWZ00190
IF(Y(I,2,K).NE.0.) GO TO 5700 CWZ00200
Q1(I,1,K)=1. CWZ00210
Q1(I,2,K)=1. CWZ00220
GO TO 5900 CWZ00230
5700 CONTINUE CWZ00240
C* CALCULATE INITIAL VOLLEY FOR HC SMOKE CWZ00250
IF(Y(I,1,K).GT.Y(I,2,K)) Y(I,1,K)=Y(I,2,K) CWZ00260
Q1(I,1,K)=X0/Y(I,1,K) CWZ00270
Q5=AIN(T(Q1(I,1,K))) CWZ00280
Q6=Q1(I,1,K)-Q5 CWZ00290
IF(Q6.EQ.0.) GO TO 5800 CWZ00300
Q1(I,1,K)=Q5+1. CWZ00310
5800 CONTINUE CWZ00320
C* CALCULATE NUMBER OF GUNS FOR SUSTAINING VOLLEYS (HC) CWZ00330
Q1(I,2,K)=X0/Y(I,2,K) CWZ00340
Q5=AIN(T(Q1(I,2,K))) CWZ00350
Q6=Q1(I,2,K)-Q5 CWZ00360
IF(Q6.EQ.0.) GO TO 5900 CWZ00370
Q1(I,2,K)=Q5+1. CWZ00380
5900 CONTINUE CWZ00390
C* CALCULATE RATE OF FIRE FOR HC SMOKE CWZ00400
R(I,1)=0.5 CWZ00410
IF(C(I,1).NE.0.) GO TO 6000 CWZ00420
R(I,1)=0. CWZ00430
6000 CONTINUE CWZ00440
C* CALCULATE TOTAL NUMBER OF ROUNDS REQUIRED (HC SMOKE) CWZ00450
P(I,K,1)=Q1(I,1,K)*(0.5*Q1(I,2,K)*(TIME-2.)) CWZ00460
Q5=AIN(T(P(I,K,1))) CWZ00470
Q6=P(I,K,1)-Q5 CWZ00480
IF(Q6.EQ.0.) GO TO 6100 CWZ00490
P(I,K,1)=Q5+1. CWZ00500
6100 CONTINUE CWZ00510
IF(I.LT.3.AND.IP0.GT.4) GO TO 6775 CWZ00520
C* SHELL SPACING (Z) AND VOLLEYS (Q) - WP SMOKE CWZ00530
IF(C(I,2).NE.0.) GO TO 6200 CWZ00540
Z(I,1,K)=0. CWZ00550
Z(I,2,K)=0. CWZ00560
Q2(I,1,K)=0. CWZ00570
Q2(I,2,K)=0. CWZ00580
GO TO 6400 CWZ00590
6200 CONTINUE CWZ00600
IF(I.LT.3) GO TO 6250 CWZ00610
IF(I.GT.2) Q2(I,1,K)=0.6*C(I,2)/V(K) CWZ00620
GO TO 6300 CWZ00630
6250 CONTINUE CWZ00640
Z(I,1,K)=V(K)/C(I,2)*100. CWZ00650
Z(I,2,K)=Z(I,1,K) CWZ00660
Q2(I,1,K)=X0/Z(I,2,K)+1. CWZ00670
6300 CONTINUE CWZ00680
Q5=AIN(T(Q2(I,1,K))) CWZ00690
Q6=Q2(I,1,K)-Q5 CWZ00700

```


IF (Q6 .EQ. 0.) GO TO 6350	CWZ00710
Q2(I,1,K)=Q5+1.	CWZ00720
6350 CONTINUE	CWZ00730
Q2(I,2,K)=Q2(I,1,K)	CWZ00740
6400 CONTINUE	CWZ00750
C* RATE OF FIRE FOR WP SMOKE	CWZ00760
IF (C(I,2) .NE. 0.) GO TO 6425	CWZ00770
R(I,2)=0.	CWZ00780
GO TO 6600	CWZ00790
6425 CONTINUE	CWZ00800
IF (I .GT. 2) GOTO 6450	CWZ00810
R(I,2)=(Z(I,2,K)+60.)/S3	CWZ00820
GO TO 6475	CWZ00830
6450 CONTINUE	CWZ00840
R(I,2)=120./S3	CWZ00850
6475 CONTINUE	CWZ00860
R(I,2)=R(I,2)/20.	CWZ00870
R5=AINTR(R(I,2))	CWZ00880
R6=R(I,2)-R5	CWZ00890
IF (R6 .LT. 0.5) GO TO 6500	CWZ00900
R5=R5+1.	CWZ00910
6500 CONTINUE	CWZ00920
IF (R5 .NE. 0.) GO TO 6550	CWZ00930
R5=1.	CWZ00940
6550 CONTINUE	CWZ00950
R(I,2)=R5*20./60.	CWZ00960
6600 CONTINUE	CWZ00970
C* CALCULATE TOTAL NUMBER OF ROUNDS REQUIRED (WP)	CWZ00980
IF (C(I,2) .NE. 0.) GO TO 6650	CWZ00990
P(I,K,2)=0.	CWZ01000
GO TO 6700	CWZ01010
6650 CONTINUE	CWZ01020
IF (I .GT. 2) GO TO 6700	CWZ01030
P(I,K,2)=Q2(I,2,K)*(1./R(I,2))*(TIME-R(I,2))	CWZ01040
GO TO 6750	CWZ01050
6700 CONTINUE	CWZ01060
P(I,K,2)=Q2(I,1,K)*(X0/60.+1.)*(1./R(I,2))*(TIME-R(I,2))	CWZ01070
6750 CONTINUE	CWZ01080
Q5=AINTP(P(I,K,2))	CWZ01090
Q6=P(I,K,2)-Q5	CWZ01100
IF (Q6 .EQ. 0.) GO TO 6800	CWZ01110
P(I,K,2)=Q5+1.	CWZ01120
GO TO 6800	CWZ01130
6775 CONTINUE	CWZ01140
C* CALCULATIONS FOR E AND F STABILITY CAT (STABLE FLOW)	CWZ01150
C* INITIAL SHELL SPACING FOR WP SMOKE	CWZ01160
IF (I .EQ. 1 .AND. K .EQ. 1) Z(I,1,K)=100.	CWZ01170
IF (I .EQ. 2 .AND. K .EQ. 1) Z(I,1,K)=50.	CWZ01180
IF (I .LT. 3 .AND. K .EQ. 2) Z(I,1,K)=100.	CWZ01190
C* SUSTAINING SHELL SPACING FOR WP SMOKE	CWZ01200
IF (I .EQ. 1 .AND. K .EQ. 1) Z(I,2,K)=100.	CWZ01210
IF (I .EQ. 1 .AND. K .EQ. 2) Z(I,2,K)=200.	CWZ01220
IF (I .EQ. 2 .AND. K .EQ. 1) Z(I,2,K)=50.	CWZ01230
IF (I .EQ. 2 .AND. K .EQ. 2) Z(I,2,K)=100.	CWZ01240
C* INITIAL VOLLEY FOR WP SMOKE	CWZ01250
Q2(I,1,K)=X0/Z(I,1,K)+1.	CWZ01260
C* SUSTAINING VOLLEY FOR WP SMOKE	CWZ01270
Q2(I,2,K)=X0/Z(I,2,K)+1.	CWZ01280
C* RATE OF FIRE FOR WP SMOKE	CWZ01290
IF (K .EQ. 1) R(I,2)=.5	CWZ01300
IF (K .EQ. 2) R(I,2)=1.	CWZ01310
C* TOTAL NUMBER OF ROUNDS REQUIRED (WP)	CWZ01320
P(I,K,2)=Q2(I,1,K)+Q2(I,2,K)*1./R(I,2)*(TIME-R(I,2))	CWZ01330
6800 CONTINUE	CWZ01340
R(I,2)=1./R(I,2)	CWZ01350
6900 CONTINUE	CWZ01360
RETURN	CWZ01370
END	CWZ01380

SUBROUTINE ITAM(IERR,TFNL)

THIS ROUTINE IS AN INVERSION OF THE NV&EOL TARGET ACQUISITION MODEL, WITH EMPHASIS ON THE DEGRADATION (TRANSMITTANCE) REQUIRED TO PREVENT DETECTION ABOVE A GIVEN PROBABILITY.

INPUTS ARE ON STANDARDIZED RECORDS:

KEY WORD - COLUMNS 1-4, FROM AMONG TARV,SENS,GO,PAGE AND DONE.
DATA FIELDS, ALL REAL - COLS 11-20, 21-30, 31-40,...,71-80.

INPUT CARDS ARE ORDER INDEPENDENT, WITH ARBITRARY OR SYSTEM DEFAULTS CHOSEN IF CARDS ARE LEFT OUT, OR IF VALUES ARE INPUT 0. THE EXCEPTION TO ORDER INDEPENDENCE IS THAT AFTER EACH SET OF INPUTS A (GO) CARD MUST BE PLACED TO INITIATE EXECUTION OF ONE LOOP THROUGH THE PROGRAM, FOLLOWING, OR IN PLACE OF THE LAST INPUT-SET GO CARD, A (DONE) CARD MUST BE PROVIDED TO CAUSE CONTROL TO EXIT ITAM AND RETURN TO THE SCREEN EXEC MODULE.

IN SUBSEQUENT RUNS (DELINEATED BY GO CARDS) TABLES MAY BE PRODUCED LINE-BY-LINE, AND ANY INPUT WHICH HAS NOT BEEN CHANGED (IE INPUT AS 0.) WILL USE THE VALUE GIVEN ON THE PREVIOUS RUN. EXCEPTIONS ARE THE FOV AND AMAG VALUES WHICH MUST BE SPECIFIED WHENEVER A NEW DEVICE NUMBER (LSC) IS INPUT, OR THESE VALUES ASSUME DEFAULTS FOR THE DEVICE.

KEY WORD	COLS.	VARIABLE	DESCRIPTION
TARV	1-4	ACON	TARGET/SCENARIO DESCRIPTION
	11-20		- INTRINSIC CONTRAST (DIMENSION-LESS) OR TEMPERATURE DIFFERENCE OF TARGET/BACKGROUND (DEG K) FOR THERMAL DEVICES.
	21-30	SOG	- SKY/GROUND RATIO, (DIMENSION-LESS), USED FOR NON-THERMAL DEVICES. SOG SHOULD INCLUDE A FACTOR FOR CLOUD REFLECTANCE
	31-40	R	- RANGE TO TARGET (KM)
	41-50	TAR SZ	- TARGET MINIMUM DIMENSION (M).
	51-60	ZONE	- SEARCH ZONE (DEG.**2)
	61-70	ALFLG	- AMBIENT ILLUM. CATEGORY (SEE LIST BELOW) CONVERTED TO NAL, AN INTEGER, IN PROGRAM. USED FOR NON-THERMAL DEVICES.
	71-80	ALIGHT	- IF ALFLG=0., THE USER MUST PROVIDE AN AMBIENT ILLUM. HERE (FT. COLS.)
SENS	1-4	PS	SENSOR DESCRIPTION
	11-20		- INPUT PROBABILITY OF DETECTION. (DIMENSIONLESS)
	21-30	DVNUM	- DEVICE NUMBER. CONVERTED TO LSC, AN INTEGER, IN PROGRAM.
	31-40	DMODE	- OPERATIONAL MODE (1. = WIDE FOV, 2. = NARROW FOV). CONVERTED TO MODE, AN INTEGER, IN PROGRAM.
	41-50	FOV	- FIELD OF VIEW (DEG.)
	51-60	AMAG	- MAGNIFICATION (FOR VISIBLE ONLY)
	61-70	AJOB	- LEVEL OF ACQUISITION (MEDIAN RESOLVABLE NORMALIZED CYCLES, USUALLY 1. FOR DETECTION)
PAGE	1-4		(OPTIONAL) - FORCE PAGE EJECT, WRITE NEW HEADER. (USEFUL FOR TABLE GENERATION.)
GO	1-4		EXECUTE ONE LOOP WITH GIVEN INPUTS
DONE	1-4		END COMPUTATIONS AND EXIT THE ITAM

ITA00010
ITA00020
ITA00030
ITA00040
ITA00050
ITA00060
ITA00070
ITA00080
ITA00090
ITA00100
ITA00110
ITA00120
ITA00130
ITA00140
ITA00150
ITA00160
ITA00170
ITA00180
ITA00190
ITA00200
ITA00210
ITA00220
ITA00230
ITA00240
ITA00250
ITA00260
ITA00270
ITA00280
ITA00290
ITA00300
ITA00310
ITA00320
ITA00330
ITA00340
ITA00350
ITA00360
ITA00370
ITA00380
ITA00390
ITA00400
ITA00410
ITA00420
ITA00430
ITA00440
ITA00450
ITA00460
ITA00470
ITA00480
ITA00490
ITA00500
ITA00510
ITA00520
ITA00530
ITA00540
ITA00550
ITA00560
ITA00570
ITA00580
ITA00590
ITA00600
ITA00610
ITA00620
ITA00630
ITA00640
ITA00650
ITA00660
ITA00670
ITA00680
ITA00690
ITA00700

```
ROUTINE. (MAY BE USED IN PLACE OF ITA00710
OR FOLLOWING THE LAST GO CARD) ITA00720
ITAO0730
OUTPUTS: TFNL, THE TOTAL PATH TRANSMITTANCE (DIMENSIONLESS) ITA00740
REQUIRED TO KEEP THE PROBABILITY OF DETECTION LESS ITA00750
THAN OR EQUAL TO THAT INPUT, AS COMPUTED FROM THE ITA00760
FINAL SET OF INPUTS PRECEDING THE DONE CARD. ITA00770
```

```
PRINT OUTPUTS: ALL INPUT PARAMETERS AND
C      = CONTRAST OR TEMPERATURE DIFFERENCE AT DEVICE
        REQUIRED FOR GIVEN LEVEL OF DETECTION PROBABILITY
RC     = RESOLVABLE CYCLES REQUIRED AT TARGET FOR GIVEN
        PROBABILITY OF DETECTION.
TTOT   = TOTAL PATH TRANSMITTANCE TO REDUCE ACON AT TARGET
        TO C AT DEVICE.
```

SUBROUTINES: ITAM CALLED BY SCREEN. ITAM CALLS CYCLE, CINY, TREQ.

AMBIENT ILLUMINATION CATEGORIES:

ALFLG OR NAL	FT. CDLS.	APPROX. CONDITIONS
0	USER SPEC.	
1	100.	CLEAR TO SLIGHT OVERCAST
2	10.	HEAVILY OVERCAST DAY
3	1.	SUNSET
4	0.1	1/4 HR. AFTER SUNSET
5	0.01	1/2 HR. AFTER SUNSET
6	0.001	MOONLIT, CLEAR
7	0.0001	MOONLIT, OVERCAST
8	0.00001	STARLIGHT ONLY
9	0.000001	MINIMAL STARLIGHT

```

DIMENSION RV(7),IT(2),AMAGW(14),AMAGN(14),FOVW(14),FOVN(14),
*ALFA(9),IR(10),BETA(9),GAMA(9),AMRAL(6,3),AMRGM(6,3)
COMMON /IUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
COMMON /GEOMET/PTS(15),IGEOSW
DATA IR/2HDO,2HNE,2HGO,2H ,2HTA,2HRV,2HSE,2HNS,2HPA,2HGE/
DATA AMAGW /1.0,7.0,12*0./,
* AMAGN /1.0,7.0,12*0./,
* FOVW /24.5,8.0,15.0,9.0,6.0,4.67,4.67,10.62,9.24,8.0,4*0./,
* FOVN /24.5,8.0,3*0.,1.56,0.,3.54,2.31,1.0,4*0./
DATA TGT,SGR,FV,RN,ALV,DMG,AJB,PD,DVN,DMD,ACN /2.3,
* 3.,7.,1.,1.,1.,1.,1.,1.,1.,1.,1.,1./
DATA ALFA /1.089005,.009730,1.8528,-0.0002197,.139,0.,0.,0.,0./
DATA BETA /-1.801654,.03536,-3.13169,.0514,0.,.0651,.1124,.0968,
* .12/
DATA GAMA/4.217775,2.3575,5.17227,2.32178,2.7933,1.6149,1.3364,
* .56867,.29/
DATA AMRAL/.05850156,.1022697,.2444688,.7264854,3.736508,
* -16.87941,.06709132,.1137755,.3177794,7642010,6.549327,
* -16.18440,.04109820,.0715830,.1524969,.2319469,1.494814,
* 5.731956/
DATA AMRGM/2.94273,2.43789,2.11113,2.05233,3.33957,-4.98529,
* 5.22111,4.28211,4.13668,3.44685,9.16003,-7.70891,
* 2.80725,2.70329,2.48447,1.63044,2.73650,3.47838 /
** INITIALIZE VALUES (FIRST EXECUTION)
PB=.632
ACON=0.
SOG=0.
RNG=0.
TARSZ=0.
UZONE=0.
ALFLG=0.
ALIGHT=0.
PDET=0.
DVNUM=0.
DMODE=0.
DFOV=0.

```

DMAG=0.	ITA01410
AJOB=0.	ITA01420
TBAR=0.	ITA01430
NAL=0	ITA01440
TFNL=1.	ITA01450
ILAST=0	ITA01460
IFLGTV=0	ITA01470
IFLCMG=0	ITA01480
C ***** OUTPUT LINE COUNTER	ITA01490
ILINE=0	ITA01500
C ***** INITIALIZE VALUES (REPEATED EXECUTIONS)	ITA01510
C ***** INPUT CARD COUNTER	ITA01520
4 ICOV=0	ITA01530
IF (ILAST.NE.0) GOTO 6	ITA01540
C ***** FLAG TO WRITE TITLE	ITA01550
ITITL=0	ITA01560
ITOT=1.	ITA01570
RC=0.	ITA01580
C=0.	ITA01590
ICM=0	ITA01600
INEWD=0	ITA01610
C ***** READ A CARD	ITA01620
5 READ(IQIN,900) IT(1),IT(2),(RV(I),I=1,7)	ITA01630
900 FORMAT(2A2,6X,7F10.2)	ITA01640
ICOV=ICOV+1	ITA01650
IF(IT(1).EQ.IR(1).AND.IT(2).EQ.IR(2)) GO TO 6	ITA01660
IF(IT(1).EQ.IR(3).AND.IT(2).EQ.IR(4)) GO TO 3	ITA01670
IF(IT(1).EQ.IR(5).AND.IT(2).EQ.IR(6)) GO TO 7	ITA01680
IF(IT(1).EQ.IR(7).AND.IT(2).EQ.IR(8)) GO TO 8	ITA01690
IF(IT(1).EQ.IR(9).AND.IT(2).EQ.IR(10)) GO TO 2	ITA01700
WRITE(IQOUT,901) IT(1),IT(2),(RV(I),I=1,7)	ITA01710
901 FORMAT(1X,61HTHE FOLLOWING CARD DOES NOT CONFORM TO ITAM INPUT CON	ITA01720
*VENTIONS/1X,2A2,6X,7E10.3)	ITA01730
IF(ICOV.LE.5) GO TO 5	ITA01740
IERR=1	ITA01750
GOTO 1	ITA01760
C ***** ALL DONE	ITA01770
6 IF (ICOV.GT.1) GOTO 3	ITA01780
1 WRITE(IQOUT,902) TFNL	ITA01790
902 FORMAT(1H0,5X,41H*** FINAL TOTAL TRANSMISSION FROM ITAM = ,F5.3)	ITA01800
RETURN	ITA01810
C ***** TARGET CARD PROCESSING (TARV)	ITA01820
7 RCHK=ACON	ITA01830
IF(RV(1).NE.0.) ACON=RV(1)	ITA01840
IF(ACON.NE.RCHK) ITITL=1	ITA01850
IF(RV(2).NE.0.) SOG=RV(2)	ITA01860
IF(RV(3).NE.0.) RNG=RV(3)	ITA01870
IF(IGEOSW.NE.1) GO TO 477	ITA01880
RNG=SQRT((PTS(1)-PTS(4))**2+(PTS(2)-PTS(5))**2+(PTS(3)-PTS(6))**2)	ITA01890
477 CONTINUE	ITA01900
RCHK=TARSZ	ITA01910
IF(RV(4).NE.0.) TARSZ=RV(4)	ITA01920
IF(TARSZ.NE.RCHK) ITITL=1	ITA01930
RCHK=UZONE	ITA01940
IF(RV(5).NE.0.) UZONE=RV(5)	ITA01950
IF(RCHK.NE.UZONE) ITITL=1	ITA01960
IF(RV(6).NE.0.) ALFLG=RV(6)	ITA01970
IF(RV(7).NE.0.) ALIGHT=RV(7)	ITA01980
GO TO 5	ITA01990
C ***** SENSOR CARD PROCESSING (SENS)	ITA02000
8 IF(RV(1).GT.0.) PDET=RV(1)	ITA02010
RCHK=DVNUM	ITA02020
IF(RV(2).NE.0.) DVNUM=RV(2)	ITA02030
IF(DVNUM.NE.RCHK) ITITL=1	ITA02040
IF(DVNUM.NE.RCHK) INEWD=1	ITA02050
RCHK=DMODE	ITA02060
IF(RV(3).NE.0.) DMODE=RV(3)	ITA02070
IF(DMODE.NE.RCHK) ITITL=1	ITA02080
IF(INEWD.EQ.1) DFOV=0.	ITA02090
IF(INEWD.EQ.1) IFLGTV=0	ITA02100

	RCHK=DFOV	ITA02110
	IF(RV(4).NE.0.) DFOV=RV(4)	ITA02120
	IF(RV(4).NE.0..AND.IFLGFV.EQ.1) IFLGFV=0	ITA02130
	IF(DFOV.NE.RCHK) ITITL=1	ITA02140
	IF(INEWD.EQ.1) DMAG=0.	ITA02150
	IF(INEWD.EQ.1) IFLGMG=0	ITA02160
	RCHK=DMAG	ITA02170
	IF(RV(5).NE.0.) DMAG=RV(5)	ITA02180
	IF(RV(5).NE.0..AND.IFLGMG.EQ.1) IFLGMG=0	ITA02190
	IF(DMAG.NE.RCHK) ITITL=1	ITA02200
	RCHK=AJOB	ITA02210
	IF(RV(6).NE.0.) AJOB=RV(6)	ITA02220
	IF(AJOB.NE.RCHK) ITITL=1	ITA02230
	GO TO 5	ITA02240
C	***** PAGE TO NEW HEADER	ITA02250
2	ITITL=1	ITA02260
	GOTO 5	ITA02270
C	***** BEGIN PROCESSING (GO)	ITA02280
3	***** FIRST CHECK IF DEFAULTS NEEDED, AND SET INTEGER VALUES, LIGHT CA	ITA02290
5	ILAST=1	ITA02300
	RCHK=DVNUM	ITA02310
	IF(DVNUM.EQ.0.) DVNUM=DVN	ITA02320
	IF(RCHK.NE.DVNUM) ITITL=1	ITA02330
	LSC=IFIX(DVNUM+.0001)	ITA02340
	IF(LSC.LT.1) LSC=1	ITA02350
	IF(LSC.GT.14) LSC=14	ITA02360
	RCHK=DMODE	ITA02370
	IF(DMODE.EQ.0.) DMODE=DMD	ITA02380
	IF(RCHK.NE.DMODE) ITITL=1	ITA02390
	MODE=IFIX(DMODE+.0001)	ITA02400
	IF(MODE.LT.1) MODE=1	ITA02410
	IF(MODE.GT.2) MODE=2	ITA02420
	IF(PDET.EQ.0.) PDET=PD	ITA02430
	PS=PDET	ITA02440
	IF(PS.GT.1.) PS=1.	ITA02450
	IF(PS.LT.0.) PS=0.	ITA02460
	RCHK=DFOV	ITA02470
	IF(DFOV.LE.0.0.AND.MODE.EQ.1) DFOV=FOVW(LSC)	ITA02480
	IF(DFOV.LE.0.0.AND.MODE.EQ.2) DFOV=FOVN(LSC)	ITA02490
	IF(DFOV.LE.0.0) IFLGFV=1	ITA02500
	IF(DFOV.LE.0..AND.MODE.EQ.1) DFOV=FV	ITA02510
	IF(DFOV.LE.0..AND.MODE.EQ.2) DFOV=FV/2.	ITA02520
	IF(RCHK.NE.DFOV) ITITL=1	ITA02530
	FOV=DFOV	ITA02540
	RCHK=DMAG	ITA02550
	IF(DMAG.LE.0.0.AND.MODE.EQ.1) DMAG=AMAGW(LSC)	ITA02560
	IF(DMAG.LE.0.0.AND.MODE.EQ.2) DMAG=AMAGN(LSC)	ITA02570
	IF(DMAG.LE.0.0) IFLGMG=1	ITA02580
	IF(DMAG.LE.0.0) DMAG=DMG	ITA02590
	IF(RCHK.NE.DMAG) ITITL=1	ITA02600
	AMAG=DMAG	ITA02610
	RCHK=AJOB	ITA02620
	IF(AJOB.EQ.0.0) AJOB=AJB	ITA02630
	IF(RCHK.NE.AJOB) ITITL=1	ITA02640
	RCHK=ACON	ITA02650
	IF(ACON.EQ.0.0) ACON=ACN	ITA02660
	IF(RCHK.NE.ACON) ITITL=1	ITA02670
	C=ACON	ITA02680
	IF(SOG.EQ.0.0) SOG=SGR	ITA02690
	IF(RNG.LE.0.0) RNG=RN	ITA02700
	R=RNG	ITA02710
	RCHK=TARSZ	ITA02720
	IF(TARSZ.LE.0.0) TARSZ=TGT	ITA02730
	IF(RCHK.NE.TARSZ) ITITL=1	ITA02740
	DIM=TARSZ	ITA02750
	RCHK=UZONE	ITA02760
	IF(UZONE.LE.0.0) UZONE=FOV**2	ITA02770
	IF(RCHK.NE.UZONE) ITITL=1	ITA02780
	ZONE=UZONE	ITA02790
	IF(ALFLG.LT.0.) ALFLG=0.	ITA02800

```

IF (ALFLG.GT.9.) ALFLG=9.
IAL=IFIX(ALFLG+0.0001)
IF (IAL.GT.0) AL=10.**((3-IAL)+1.E-10)
IF (IAL.EQ.0.AND.ALIGHT.EQ.0.0) ALIGHT=ALV
IF (IAL.EQ.0) AL=ALIGHT
C ***** CHECK AND WRITE PAGE TITLE HEADING
10 ILINE=ILINE+1
IF (ITITL.EQ.0.AND.ILINE.LE.49) GO TO 11
IF (ITITL.EQ.0.AND.ILINE.GT.49) WRITE (IOOUT,905)
905 FORMAT(1H,35X,28H** CONTINUED ON NEXT PAGE **)
WRITE(IOOUT,910)
910 FORMAT(1H0,41X,43(1H*)/41X,1H*,41X,1H*/41X,1H*,2X,37HINVERSE S
*TATIC TARGET DETECTION MODEL,2X,1H*/41X,1H*,41X,1H*/41X,43(1H*))
WRITE(IOOUT,911)
911 FORMAT(1H0,64X,28HTARGET INTRINSIC CONTRAST OR)
WRITE(IOOUT,912) LSC,ACON
912 FORMAT(20X,13HDEVICE NUMBER,4X,12,32X,22HTEMPERATURE DIFFERENCE,
* 2X,F7.3)
IF (MODE.EQ.2) WRITE(IOOUT,913) DIM
913 FORMAT(1H0,19X,11HFOV TYPE - ,4X,6HNARROW,24X,28HMINIMUM TARGET DI
* MENSION (M),2X,F7.3)
IF (MODE.EQ.1) WRITE(IOOUT,914) DIM
914 FORMAT(1H0,19X,11HFOV TYPE - ,6X,4HWIDE,24X,28HMINIMUM TARGET DIME
* NSION (M),2X,F7.3)
IF (IFLGFV.EQ.0) WRITE(IOOUT,915) FOV,AJOB
915 FORMAT(1H0,19X,9HFOV (DEG),5X,F7.3,24X,27HACQUISITION LEVEL (50 PC
* NT),3X,F7.3)
IF (IFLGFV.EQ.1) WRITE(IOOUT,916) FOV,AJOB
916 FORMAT(1H0,19X,9HFOV (DEG),5X,F7.3,1X,19H(ARBITRARY DEFAULT),
* 4X,27HACQUISITION LEVEL (50 PCNT),3X,F7.3)
IF (IFLGMG.EQ.0) WRITE(IOOUT,917) AMAG,ZONE
917 FORMAT(1H0,19X,13HMAGNIFICATION,2X,F6.3,24X,24HSEARCH ZONE (DEGREE
* S**2),6X,F7.3)
IF (IFLGMG.EQ.1) WRITE(IOOUT,918) AMAG,ZONE
918 FORMAT(1H0,19X,13HMAGNIFICATION,2X,F6.3,1X,19H(ARBITRARY DEFAULT),
* 4X,24HSEARCH ZONE (DEGREES**2),6X,F7.3)
WRITE (IOOUT,920)
920 FORMAT(1H0,52X,35HREQUIRES (TO DEFEAT DEVICE) AT MOST/
* 4X,14HFOR NO GREATER,6X,5HUNDER,8X,3HAND,7X,2HAT,4X,
* 35(1H-))
WRITE(IOOUT,919)
919 FORMAT(8X,5HINPUT,11X,5HINPUT,7X,5HINPUT,5X,5HINPUT,3X,
* 8HCOMPUTED,2X,11HCONTRAST OR,3X,8HCOMPUTED/6X,
* 9HDETECTION,6X,11HAMB, ILLUM,2X,10HISKY/GROUND,2X,
* 5HRANGE,2X,10HRESOLVABLE,1X,11HTEMP. DIFF.,2X,10HTOTAL PATH
* /3X,5HPROB.,2X,9HTIME(SEC),3X,9H(FT CDLS),5X,5HRATIO,5X,
* 4H(KM),3X,10HCYCLES, RC,2X,9HAT DEVICE,2X,13HTRANSMITTANCE,
* 13X,8HCOMMENTS/3X,5(1H-),2X,9(1H-),2X,11(1H-),2X,10(1H-),
* 1X,6(1H-),2X,10(1H-),2X,9(1H-),2X,13(1H-),2X,30(1H-))
ILINE=23
C ***** BEGIN COMPUTATIONS FOV,RC,TBAR
11 S=DIM/R
IF (FOV.LE.0.) FOV=.0001
TS=1.7*ZONE/FOV**2
IF (ZONE.GT.9.0.AND.FOV.GT.5.) TS=(1.7*ZONE)/(5.0*FOV)
IF (TS.LT.5) TS=.5
CALL CYCLE(PS,PB,AJOB,RC)
IF (AL.LE.0.) AL=1.E-7
ALPRT=AL
IF (LSC.EQ.13) GOTO 13
TBAR=0.
IF (RC.LT.0.1) GOTO 14
IF ((LSC.GT.5.AND.LSC.LT.10).OR.LSC.EQ.11.OR.LSC.EQ.14) GO TO 12
RCS=RC
PINF=1.-EXP(-1.7*RCS/6.8)
IF (PINF.LE.0.) PINF=.0001
TBAR=0.5*TS*(2.-PINF)/PINF
GO TO 14
12 TBAR=0.5*TS*(2.0-PS)/PS
GO TO 14

```

```

C ***** DEVICE #13
13  NAL=1
    ICM=0
    PS=.99
    TBAR=1.8
    C=ACON
    TTOT=1.
    TFNL=TTOT
    GO TO 22
C ***** COMPUTE C BASED ON AMBIENT ILLUMINATION, INTERPOLATE IF NECESSARY
14  IF (S.LE.0.) S=.0001
    IF (RC.LT.0.) RC=0.
    RC=RC/S
    IF (AMAG.LE.0.) AMAG=.0001
    IF (LSC.EQ.1.OR.LSC.EQ.2) RC=RC/AMAG
    IF (LSC.EQ.2) AL=0.7*AL
C ***** CORRECT FOR FIELD OF VIEW
    IF (LSC.LT.6.OR.LSC.EQ.7) GOTO 140
    IF (LSC.EQ.6).AND.(MODE.EQ.2) RC=RC/3.
    IF (LSC.EQ.8).AND.(MODE.EQ.1) RC=RC*3.
    IF (LSC.EQ.9).AND.(MODE.EQ.1) RC=RC*4.
    IF (LSC.EQ.11).AND.(MODE.EQ.1) RC=RC*3.
    IF (LSC.EQ.14).AND.(MODE.EQ.1) RC=RC*3.
    IF (LSC.EQ.10).AND.(MODE.EQ.1) RC=RC*8.
    IF (LSC.EQ.12).AND.(MODE.EQ.2) RC=RC/4.
140  CONTINUE
    IF (LSC.EQ.2) IAL=0
    IF (IAL.GT.0) GO TO 20
    AV=100.
    IF (AL.GT.100.) AL=100.
    DO 15 I=1,9
    IF (AL.GT.(0.9**I*AV)) GO TO 16
    AV=AV/10.
15  CONTINUE
    I=2
    IF (LSC.GT.5) GOTO 160
    NAL=0
    ICM=0
    GO TO 22
C ***** CHECK IF INTERPOLATION NEEDED, IF NOT, GO TO 20
16  IF (LSC.GT.5) GOTO 160
    IF (AL.GT.(1.001*AV)) GO TO 17
160  IAL=1
    GO TO 20
C ***** INTERPOLATE
17  NAL1=I-1
    NAL2=I
    ICM1=0
    ICM2=0
    CALL INTAL(LSC,RC,C,AL,NAL1,NAL2,ICM1,ICM2,ALFA,BETA,GAMA,AMRAL,
    *AMRGM)
    NAL=0
    IF (NAL1.EQ.0.AND.NAL2.EQ.0) GO TO 22
    NAL=NAL2
    IF (NAL1.GT.0) GO TO 18
    ICM=ICM2
    GO TO 21
18  IF (NAL2.GT.0) GO TO 19
    NAL=NAL1
    ICM=ICM1
    GO TO 21
19  IF (ICM1.GT.0.AND.ICM2.GT.0) ICM=1
    IF (ICM1.LT.0.AND.ICM2.LT.0) ICM=-1
    GO TO 21
C ***** NO INTERPOLATION NEEDED
20  NAL=IAL
    CALL CINV(LSC,RC,NAL,C,ICM,ALFA,BETA,GAMA,AMRAL,AMRGM)
21  IF (ICM.EQ.-1) C=.99*C
    CALL TREQ(ACON,SOC,C,LSC,TTOT)
    IF (ICM.EQ.1) TTOT=1.

```

```

ITA03510
ITA03520
ITA03530
ITA03540
ITA03550
ITA03560
ITA03570
ITA03580
ITA03590
ITA03600
ITA03610
ITA03620
ITA03630
ITA03640
ITA03650
ITA03660
ITA03670
ITA03680
ITA03690
ITA03700
ITA03710
ITA03720
ITA03730
ITA03740
ITA03750
ITA03760
ITA03770
ITA03780
ITA03790
ITA03800
ITA03810
ITA03820
ITA03830
ITA03840
ITA03850
ITA03860
ITA03870
ITA03880
ITA03890
ITA03900
ITA03910
ITA03920
ITA03930
ITA03940
ITA03950
ITA03960
ITA03970
ITA03980
ITA03990
ITA04000
ITA04010
ITA04020
ITA04030
ITA04050
ITA04040
ITA04060
ITA04070
ITA04080
ITA04090
ITA04100
ITA04110
ITA04120
ITA04130
ITA04140
ITA04150
ITA04160
ITA04170
ITA04180
ITA04190
ITA04200

```

	TFNL=TTOT	ITA04210
C *****	PRINT RESULTS	ITA04220
22	IF(NAL.GT.0) GO TO 23	ITA04230
	TFNL=1	ITA04240
	WRITE(100UT,930) PS,TBAR,ALPRNT,SQG,R,TFNL	ITA04250
930	FORMAT(3X,F5.3,F9.2,F14.6,F10.3,F9.2,4X,6(1H-),7X,5(1H-),5X,	ITA04260
*	F8.3,6X,28HNOTE- AMBIENT ILLUM. OUTSIDE/91X,	ITA04270
*	29HDEVICE OPERATIONAL LIMITS. NO/91X,	ITA04280
*	21HOBSCURATION REQUIRED./1X)	ITA04290
	ILINE=ILINE+3	ITA04300
	GO TO 4	ITA04310
23	IF(TTOT.CE.1.,.AND.ICM.EQ.0) GOTO 24	ITA04320
	IF(ICM.EQ.0) WRITE(100UT,931) PS,TBAR,ALPRNT,SQG,R,RC,C,TTOT	ITA04330
	IF(ICM.EQ.1) WRITE(100UT,932) PS,TBAR,ALPRNT,SQG,R,RC,TTOT,C	ITA04340
	IF(ICM.EQ.1) ILINE=ILINE+6	ITA04350
	IF(ICM.EQ.-1) WRITE(100UT,933) PS,TBAR,ALPRNT,SQG,R,RC,C,TTOT	ITA04360
	IF(ICM.EQ.-1) ILINE=ILINE+7	ITA04370
	GO TO 4	ITA04380
24	WRITE(100UT,934) PS,TBAR,ALPRNT,SQG,R,RC,C,TTOT	ITA04390
934	FORMAT(3X,F5.3,F9.2,F14.6,F10.3,F9.2,F10.3,F12.3,5X,F8.3,6X,	ITA04400
*	31HNOTE- CONTRAST (OR TEMP. DIFF.)/91X,	ITA04410
*	31HREQUIRED WOULD EXCEED INTRINSIC/91X,	ITA04420
*	31HCONTRAST (TEMP. DIFF.). NO OBS-/91X,	ITA04430
*	17HCURATION REQUIRED/1X)	ITA04440
	ILINE=ILINE+3	ITA04450
	GO TO 4	ITA04460
931	FORMAT(3X,F5.3,F9.2,F14.6,F10.3,F9.2,F10.3,F12.3,5X,F8.3)	ITA04470
932	FORMAT(3X,F5.3,F9.2,F14.6,F10.3,F9.2,F10.3,7X,5(1H-),5X,F8.3,6X,	ITA04480
*	29HNOTE- DETECTION PROBABILITY /91X,	ITA04490
*	30HREQUIRES CONTRAST (OR TEMP. /91X,	ITA04500
*	29HDIFF.) AND RESOLVABLE CYCLES /91X,	ITA04510
*	27HABOVE LIMIT FOR DEVICE. NO /91X,	ITA04520
*	26HOBSCURANT REQUIRED. DEVICE/91X,	ITA04530
*	18HUPPER LIMIT IS C= ,F8.3/1X)	ITA04540
933	FORMAT(3X,F5.3,F9.2,F14.6,F10.3,F9.2,F10.3,F12.3,5X,F8.3,6X,	ITA04550
*	30HNOTE- INPUT DETECTION PROBAB- /91X,	ITA04560
*	30HILITY REQUIRES CONTRAST (OR /91X,	ITA04570
*	30HTEMP. DIFF.) BELOW THRESHOLD. /91X,	ITA04580
*	29HVALUES ASSUMED ARE 99 PERCENT/91X,	ITA04590
*	29HOF THRESHOLD. ADDITIONAL OBS-/91X,	ITA04600
*	29HCURANT WILL NOT DECREASE /91X,	ITA04610
*	30HDETECTION PROBABILITY. /1X)	ITA04620
	END	ITA04630


```

SUBROUTINE CINV(LSC,RC,NAL,C,ICM,ALFA,BETA,GAMA,AMRAL,AMRGM)      CIN00010
DIMENSION ALFA(9),BETA(9),GAMA(9),AMRAL(6,3),AMRGM(6,3)        CIN00020
COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU CIN00030
DATA IPOB,INEG /1,-1/                                           CIN00040
THIS ROUTINE COMPUTES THE REQUIRED CONTRAST AT THE DEVICE FOR A   CIN00050
GIVEN NUMBER OF RESOLVABLE CYCLES AT THE DEVICE. EQUATIONS     CIN00060
ARE INVERSIONS OF THOSE OF THE NV&EOL TARGET ACQUISITION       CIN00070
(FOR STATIC DETECTION) MODEL.                                   CIN00080
INPUTS: LSC = DEVICE NUMBER (1-14)                             CIN00090
        RC  = RESOLVABLE CYCLES (DECIMAL)                     CIN00100
        NAL = AMBIENT ILLUMINATION CATEGORY (1-9)             CIN00110
OUTPUTS: C   = CONTRAST OR TEMPERATURE DIFFERENCE(DEG K) REQUIRED CIN00120
        ICM = OPERATIONAL LIMITS FLAG (+1 EXCEEDS LIMIT, -1   CIN00130
        NAL = SET TO 0 IF TOO MUCH OR NOT ENOUGH AMBIENT ILLUM. CIN00140
        BY DEVICE FOR GIVEN RC.                               CIN00150
        ICM = OPERATIONAL LIMITS FLAG (+1 EXCEEDS LIMIT, -1   CIN00160
        BELOW LIMIT, 0 WITHIN RANGE)                         CIN00170
        NAL = SET TO 0 IF TOO MUCH OR NOT ENOUGH AMBIENT ILLUM. CIN00180
        CFUN(RX,ALF,BET,GAM)=(BET+ALF*RX)/(GAM-RX)           CIN00190
        ICM=0                                                  CIN00200
        IF(NAL.EQ.0) GO TO 9020                                CIN00210
C ***** BRANCH TO LSC                                         CIN00220
5      GO TO (10,10,30,40,50,60,60,80,90,100,1111,1200,9060,1400), LSC CIN00230
10     GO TO (110,120,130,140,150,160,170,9060,9060), NAL      CIN00240
C ***** DEVICE #1 NAL #1                                       CIN00250
110    IF(2.74.LT.RC) GO TO 9030                                CIN00260
        IF((2.133.LT.RC).AND.(RC.LE.2.74)) GO TO 112           CIN00270
        IF((.26795.LT.RC).AND.(RC.LE.2.133)) GO TO 113         CIN00280
        IF(RC.LE..26795) GO TO 114                              CIN00290
        GO TO 9000                                              CIN00300
112    IEQ=1                                                    CIN00310
        GO TO 9050                                              CIN00320
113    IEQ=2                                                    CIN00330
        GO TO 9050                                              CIN00340
114    C=0.015                                                  CIN00350
        GO TO 9015                                              CIN00360
C ***** DEVICE #1 NAL #2                                       CIN00370
120    IF(2.74.LT.RC) GO TO 9030                                CIN00380
        IF((2.133.LT.RC).AND.(RC.LE.2.74)) GO TO 122           CIN00390
        IF((.0001.LE.RC).AND.(RC.LE.2.133)) GO TO 123         CIN00400
        IF(RC.LT..0001) GO TO 124                              CIN00410
        GO TO 9000                                              CIN00420
122    IEQ=3                                                    CIN00430
        GO TO 9050                                              CIN00440
123    IEQ=4                                                    CIN00450
        GO TO 9050                                              CIN00460
124    C=0.025                                                  CIN00470
        GO TO 9015                                              CIN00480
C ***** DEVICE #1 NAL #3                                       CIN00490
130    IF(2.29.LT.RC) GO TO 131                                CIN00500
        IF((0.49585.LE.RC).AND.(RC.LE.2.29)) GO TO 132        CIN00510
        IF(RC.LT.0.49585) GO TO 133                            CIN00520
        GO TO 9000                                              CIN00530
131    C=0.6324                                                 CIN00540
        GO TO 9035                                              CIN00550
132    IEQ=5                                                    CIN00560
        GO TO 9050                                              CIN00570
133    C=0.030                                                  CIN00580
        GO TO 9015                                              CIN00590
C ***** DEVICE #1 NAL #4                                       CIN00600
140    IF(1.5219.LT.RC) GO TO 141                              CIN00610
        IF((0.313.LE.RC).AND.(RC.LE.1.5219)) GO TO 142        CIN00620
        IF(RC.LT.0.313) GO TO 143                              CIN00630
        GO TO 9000                                              CIN00640
141    C=0.70                                                   CIN00650
        GO TO 9035                                              CIN00660
142    IEQ=6                                                    CIN00670
        GO TO 9050                                              CIN00680
143    C=0.05                                                  CIN00690

```

```

GO TO 9015
C ***** DEVICE #1      NAL #5
150  IF<1.1959,LT,RC> GO TO 9030
      IF<<0.0,LE,RC>>,AND,<RC,LE,1.1959>> GO TO 152
      IF<RC,LT,0.0> GO TO 153
      GO TO 9000
152  IEQ=7
      GO TO 9050
153  C=0.0841
      GO TO 9015
C ***** DEVICE #1      NAL #6
160  IF<0.44767,LT,RC> GO TO 9030
      IF<<0.0309,LE,RC>>,AND,<RC,LE,0.44767>> GO TO 162
      IF<RC,LT,0.0309> GO TO 163
      GO TO 9000
162  IEQ=8
      GO TO 9050
163  C=0.18
      GO TO 9015
C ***** DEVICE #1      NAL #7
170  IF<0.14,LT,RC> GO TO 9030
      IF<<0.05,LE,RC>>,AND,<RC,LE,0.14>> GO TO 172
      IF<RC,LT,0.05> GO TO 173
      GO TO 9000
172  IEQ=9
      GO TO 9050
173  C=0.50
      GO TO 9015
C ***** DEVICE #3
30   GO TO<9060,9060,9060,340,350,360,370,380,390>, NAL
C ***** DEVICE #3      NAL #4
340  IF<0.7497,LE,RC,AND,RC,LT,2.941> GO TO 342
      IF<RC,GE,2.941> GOTO 9075
      IF<RC,LT,0.7497> GO TO 9040
      GO TO 9000
342  IEQ=1
      GO TO 9080
C ***** DEVICE #3      NAL #5
350  IF<0.3988,LE,RC,AND,RC,LT,2.4350> GO TO 352
      IF<RC,GE,2.435> GOTO 9075
      IF<RC,LT,0.3988> GO TO 9040
      GO TO 9000
352  IEQ=2
      GO TO 9080
C ***** DEVICE #3      NAL #6
360  IF<0.15965,LE,RC,AND,RC,LT,2.1060> GO TO 362
      IF<RC,GE,2.106> GOTO 9075
      IF<RC,LT,0.15965> GO TO 9040
      GO TO 9000
362  IEQ=3
      GO TO 9080
C ***** DEVICE #3      NAL #7
370  IF<0.05498,LE,RC,AND,RC,LT,2.0375> GO TO 372
      IF<RC,GE,2.0375> GOTO 9075
      IF<RC,LT,0.05498> GO TO 9040
      GO TO 9000
372  IEQ=4
      GO TO 9080
C ***** DEVICE #3      NAL #8
380  IF<0.442,LE,RC,AND,RC,LT,3.2190> GO TO 382
      IF<RC,GE,3.219> GOTO 9075
      IF<<RC,LT,0.442>>,AND,<RC,GT,0.>> GO TO 383
      IF<RC,LE,0.> GOTO 384
      GO TO 9000
382  IEQ=5
      GO TO 9080
383  C=0.57
      GO TO 9015
384  C=0.33
      GOTO 9015

```

```

CIN00710
CIN00720
CIN00730
CIN00740
CIN00750
CIN00760
CIN00770
CIN00780
CIN00790
CIN00800
CIN00810
CIN00820
CIN00830
CIN00840
CIN00850
CIN00860
CIN00870
CIN00880
CIN00890
CIN00900
CIN00910
CIN00920
CIN00930
CIN00940
CIN00950
CIN00960
CIN00970
CIN00980
CIN00990
CIN01000
CIN01010
CIN01020
CIN01030
CIN01040
CIN01050
CIN01060
CIN01070
CIN01080
CIN01090
CIN01100
CIN01110
CIN01120
CIN01130
CIN01140
CIN01150
CIN01160
CIN01170
CIN01180
CIN01190
CIN01200
CIN01210
CIN01220
CIN01230
CIN01240
CIN01250
CIN01260
CIN01270
CIN01280
CIN01290
CIN01300
CIN01310
CIN01320
CIN01330
CIN01340
CIN01350
CIN01360
CIN01370
CIN01380
CIN01390
CIN01400

```

```

C ***** DEVICE #3      NAL #9
390  IF (0.172.LE.RC) GO TO 392
    IF (RC.LT.0.172.AND.RC.GT.0.) GO TO 393
    IF (RC.LE.0.) GOTO 394
    GO TO 9000
392  IEQ=6
    GO TO 9080
393  C=0.57
    GO TO 9015
394  C=.33
    GOTO 9015
C ***** DEVICE #4
40   GO TO (9060,9060,9060,440,450,460,470,480,490), NAL
C ***** DEVICE #4      NAL #4
440  IF (1.199.LE.RC.AND.RC.LT.5.2180) GO TO 442
    IF (RC.GE.5.218) GOTO 9075
    IF (RC.LT.1.199) GO TO 9040
    GO TO 9000
442  IEQ=1
    GO TO 9080
C ***** DEVICE #4      NAL #5
450  IF (0.6402.LE.RC.AND.RC.LT.4.2770) GO TO 452
    IF (RC.GE.4.277) GOTO 9075
    IF (RC.LT.0.6402) GO TO 9040
    GO TO 9000
452  IEQ=2
    GO TO 9080
C ***** DEVICE #4      NAL #6
460  IF (0.2449.LE.RC.AND.RC.LT.4.1240) GO TO 462
    IF (RC.GE.4.124) GOTO 9075
    IF (RC.LT.0.2449) GO TO 9040
    GO TO 9000
462  IEQ=3
    GO TO 9080
C ***** DEVICE #4      NAL #7
470  IF (0.08791.LE.RC.AND.RC.LT.3.4200) GO TO 472
    IF (RC.GE.3.420) GOTO 9075
    IF (RC.LT.0.08791) GO TO 9040
    GO TO 9000
472  IEQ=4
    GO TO 9080
C ***** DEVICE #4      NAL #8
480  IF (0.4394.LE.RC.AND.RC.LT.8.5970) GO TO 482
    IF (RC.GE.8.597) GOTO 9075
    IF (RC.LT.0.4394.AND.(RC.GT.0.)) GO TO 483
    IF (RC.LE.0.) GOTO 484
    GO TO 9000
482  IEQ=5
    GO TO 9080
483  C=0.33
    GO TO 9015
484  C=0.07
    GOTO 9015
C ***** DEVICE #4      NAL #9
490  IF (0.1605.LE.RC) GO TO 492
    IF (RC.LT.0.1605.AND.RC.GT.0.) GO TO 493
    IF (RC.LE.0.) GOTO 494
    GO TO 9000
492  IEQ=6
    GO TO 9080
493  C=0.33
    GO TO 9015
494  C=0.07
    GOTO 9015
C ***** DEVICE # 5
50   GO TO (9060,9060,9060,540,550,560,570,580,590), NAL
540  IF (0.9189.LE.RC.AND.RC.LT.2.8060) GO TO 542
    IF (RC.GE.2.806) GOTO 9075
    IF (RC.LT.0.9189) GO TO 9040
    GO TO 9000

```

```

CIN01410
CIN01420
CIN01430
CIN01440
CIN01450
CIN01460
CIN01470
CIN01480
CIN01490
CIN01500
CIN01510
CIN01520
CIN01530
CIN01540
CIN01550
CIN01560
CIN01570
CIN01580
CIN01590
CIN01600
CIN01610
CIN01620
CIN01630
CIN01640
CIN01650
CIN01660
CIN01670
CIN01680
CIN01690
CIN01700
CIN01710
CIN01720
CIN01730
CIN01740
CIN01750
CIN01760
CIN01770
CIN01780
CIN01790
CIN01800
CIN01810
CIN01820
CIN01830
CIN01840
CIN01850
CIN01860
CIN01870
CIN01880
CIN01890
CIN01900
CIN01910
CIN01920
CIN01930
CIN01940
CIN01950
CIN01960
CIN01970
CIN01980
CIN01990
CIN02000
CIN02010
CIN02020
CIN02030
CIN02040
CIN02050
CIN02060
CIN02070
CIN02080
CIN02090
CIN02100

```

```

542 IEQ=1
GO TO 9080
C ***** DEVICE #5 NAL #5
550 IF (0.59035.LE.RC.AND.RC.LT.2.7010) GO TO 552
IF (RC.GE.2.7010) GOTO 9075
IF (RC.LT.0.59035) GO TO 9040
GO TO 9000
552 IEQ=2
GO TO 9080
C ***** DEVICE #5 NAL #6
560 IF (0.2881.LE.RC.AND.RC.LT.2.4800) GO TO 562
IF (RC.GE.2.4800) GOTO 9075
IF (RC.LT.0.2881) GO TO 9040
GO TO 9000
562 IEQ=3
GO TO 9080
C ***** DEVICE #5 NAL #7
570 IF (0.12943.LE.RC.AND.RC.LT.1.6270) GO TO 572
IF (RC.GE.1.6270) GOTO 9075
IF (RC.LT.0.12943) GO TO 9040
GO TO 9000
572 IEQ=4
GO TO 9080
C ***** DEVICE #5 NAL #8
580 IF (0.4949.LE.RC.AND.RC.LT.2.6960) GO TO 582
IF (RC.GE.2.6960) GOTO 9075
IF (RC.LT.0.4949.AND.RC.GT.0.) GO TO 583
IF (RC.LE.0.) GOTO 584
GO TO 9000
582 IEQ=5
GO TO 9080
583 C=0.33
GO TO 9015
584 C=0.07
GO TO 9015
C ***** DEVICE #5 NAL #9
590 IF (0.1894.LE.RC.AND.RC.LT.3.2898) GO TO 592
IF (RC.GE.3.2898) GOTO 9075
IF (RC.LT.0.1894.AND.RC.GT.0.) GO TO 593
IF (RC.LE.0.) GOTO 594
GO TO 9000
592 IEQ=6
GO TO 9080
593 C=0.33
GO TO 9015
594 C=0.07
GO TO 9015
C ***** DEVICES #6 AND #7
60 IF (1.7934.LT.RC) GO TO 601
IF ((0.92376.LT.RC).AND.(RC.LE.1.7934)) GO TO 602
IF ((0.11022.LE.RC).AND.(RC.LE.0.92376)) GO TO 603
IF (RC.LT.0.11022) GO TO 9070
GO TO 9000
601 C=8.38
GO TO 9035
602 C=(.172363*RC-0.0392775)/(1.82560-RC)
GO TO 9020
603 C=(0.09772*RC)/(1.-0.34779*RC)
GO TO 9020
C ***** DEVICE #8
80 IF (4.702.LT.RC) GO TO 801
IF ((0.996.LE.RC).AND.(RC.LE.4.702)) GO TO 802
IF (RC.LT.0.996) GO TO 803
GO TO 9000
801 C=2.18
GO TO 9035
802 C=(0.0298*RC)/(1.-0.199*RC)
GO TO 9020
803 C=0.037
GO TO 9015

```

```

CIN02110
CIN02120
CIN02130
CIN02140
CIN02150
CIN02160
CIN02170
CIN02180
CIN02190
CIN02200
CIN02210
CIN02220
CIN02230
CIN02240
CIN02250
CIN02260
CIN02270
CIN02280
CIN02290
CIN02300
CIN02310
CIN02320
CIN02330
CIN02340
CIN02350
CIN02360
CIN02370
CIN02380
CIN02390
CIN02400
CIN02410
CIN02420
CIN02430
CIN02440
CIN02450
CIN02460
CIN02470
CIN02480
CIN02490
CIN02500
CIN02510
CIN02520
CIN02530
CIN02540
CIN02550
CIN02560
CIN02570
CIN02580
CIN02590
CIN02600
CIN02610
CIN02620
CIN02630
CIN02640
CIN02650
CIN02660
CIN02670
CIN02680
CIN02690
CIN02700
CIN02710
CIN02720
CIN02730
CIN02740
CIN02750
CIN02760
CIN02770
CIN02780
CIN02790
CIN02800

```

```

C ***** DEVICE #9
90 IF<0.37181,LE,RC,AND,RC,LT,9.14027> GO TO 901
  IF<RC,GE,9.14027> GOTO 9075
  IF<RC,LT,0.37181> GO TO 9070
  GO TO 9000
901 C=<0.0289*RC>/<1.-0.1092*RC>
  GO TO 9020
C ***** DEVICE #10
100 IF<NAL,GT,2> GO TO 9060
  IF<4.227,LE,RC,AND,RC,LT,91.05956> GO TO 1010
  IF<RC,GE,91.05956> GOTO 9075
  C=0.01
C NEXT STATEMENT IS LIMINAL CONTRAST ADJUSTMENT
  C=C/1.63
  GO TO 9015
1010 C=<0.0303*RC>/<13.43-0.1473*RC>
C NEXT STATEMENT IS LIMINAL CONTRAST ADJUSTMENT
  C=C/1.63
  GO TO 9020
C ***** DEVICE # 11
1111 IF<0.5057,LE,RC,AND,RC,LT,7.73353> GO TO 1110
  IF<RC,GE,7.73353> GOTO 9075
  GO TO 9070
1110 C=<0.0207*RC>/<1.-0.1291*RC>
  GO TO 9020
C ***** DEVICE # 12
1200 IF<NAL,GT,2> GO TO 9060
  IF<3.48,LE,RC,AND,RC,LT,31.48113> GO TO 1210
  IF<RC,GE,31.41883> GOTO 9075
  C=0.01
C NEXT STATEMENT IS LIMINAL CONTRAST ADJUSTMENT
  C=C/1.63
  GO TO 9015
1210 C=<0.0206*RC>/<8.06-0.2559*RC>
C NEXT STATEMENT IS LIMINAL CONTRAST ADJUSTMENT
  C=C/1.63
  GO TO 9020
C ***** DEVICE # 14
1400 IF<0.9098,LE,RC,AND,RC,LT,4.57711> GO TO 1410
  IF<RC,GE,4.57711> GOTO 9075
  C=0.037
  GO TO 9015
1410 C=<0.0297567*RC>/<0.91287-0.1991449*RC>
  GO TO 9020
9000 WRITE<100UT,9010> LSC,NAL,RC
9010 FORMAT<5X,46HBAD PARAMETER PASSED TO SUBRTN CINV **** LSC= ,I3,
  * 6H NAL= ,I3,5H RC= ,F10.4>
9015 ICM=INEG
9020 RETURN
9030 C=0.80
9035 ICM=IPOS
  GO TO 9020
9040 C=0.02
  GO TO 9015
9050 C=CFUN<RC,ALFA<IEQ>,BETA<IEQ>,GAMA<IEQ>>
  GO TO 9020
9060 NAL=0
  GO TO 9020
9070 C=0.0112
  GO TO 9015
9075 C=100.
  GOTO 9035
9080 L=LSC-2
  C=CFUN<RC,AMRAL<IEQ,L>,0.,AMRGM<IEQ,L>>
  GOTO 9020
END

```

```

CIN02810
CIN02820
CIN02830
CIN02840
CIN02850
CIN02860
CIN02870
CIN02880
CIN02890
CIN02900
CIN02910
CIN02920
CIN02930
CIN02940
CIN02950
CIN02960
CIN02970
CIN02980
CIN02990
CIN03000
CIN03010
CIN03020
CIN03030
CIN03040
CIN03050
CIN03060
CIN03070
CIN03080
CIN03090
CIN03100
CIN03110
CIN03120
CIN03130
CIN03140
CIN03150
CIN03160
CIN03170
CIN03180
CIN03190
CIN03200
CIN03210
CIN03220
CIN03230
CIN03240
CIN03250
CIN03260
CIN03270
CIN03280
CIN03290
CIN03300
CIN03310
CIN03320
CIN03330
CIN03340
CIN03350
CIN03360
CIN03370
CIN03380
CIN03390
CIN03400
CIN03410
CIN03420
CIN03430
CIN03440
CIN03450
CIN03460

```

```

SUBROUTINE INTAL(LSC,RC,C,AL,NAL1,NAL2,ICM1,ICM2,ALFA,BETA,GAMA, INT00010
*AMRAL,AMRGM) INT00020
C INTERPOLATION ON THE SURFACE IN C, RC, AMBIENT ILLUMINATION SPACE. INT00030
C THE USER INPUTS A VALUE OF RC, AND AN AMBIENT ILLUMINATION THAT INT00040
C LIES BETWEEN ONE PAIR OF MODELED RC VS C CURVES AT TWO DISCRETELY INT00050
C MODELED AMBIENT ILLUMINATIONS. THE TECHNIQUE IS COMPLICATED BY A INT00060
C REQUIREMENT TO REPRODUCE ALMOST EXACTLY THE INTERPOLATED VALUE INT00070
C FROM RC VS C OF THE NVL TARGET ACQUISITION MODEL. THUS, WHILE INT00080
C RC IS INPUT IN THIS INVERSION MODEL, THE INTERPOLATION IS BETWEEN INT00090
C RC VALUES AT CONSTANT C OVER TWO AMB. ILLUM. REGIONS. THE INT00100
C THRESHOLDS OR OPERATIONAL LIMITS OF THE DEVICE ARE TREATED SEPAR- INT00110
C ATELY. IN ALL, FOUR TYPES OF INTERPOLATION SITUATIONS ARE INT00120
C UTILIZED. INT00130
C
C DIMENSION IRGNS(21),INDEX(21),RCL(25),RCU(25),CU(25),CL(25), INT00140
*IEQ1(36),IEQ2(36),RLW(36),RUP(36),CLW(36),CUP(36),ALFA(9), INT00150
*BETA(9),GAMA(9),AMRAL(6,3),AMRGM(6,3) INT00160
COMMON /IOUNIT/IOIN,IOOUT,IPHUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU INT00170
C** NUMBER OF C REGIONS FOR INTERPOLATION CONSIDERATION, DEV. 1-5 INT00180
DATA IRGNS /4,4,3,3,2,2,1,1,1,2,1,1,1,2,1,1,1,2,1,1/ INT00190
C** STARTING INDEX FOR EACH DEVICE, AMB. ILLUM. PAIR. INT00200
DATA INDEX /1,5,9,12,15,17,19,20,21,22,24,25,26,27,28,30,31, INT00210
*32,33,34,36/ INT00220
C** LOWER LIGHT LEVEL THRESHOLDS OF RC. INT00230
DATA RCL / INT00240
*.0001, .26795, .49585, .313, .0, .0309, .05, .7497, .3988, INT00250
*.15965, .055, .442, .1742, .1, .199, .6402, .2449, .0879, .4394, INT00260
*.1605, .9189, .5903, .288, .1294, .4949, .1894 / INT00270
C** UPPER LIGHT LEVEL THRESHOLDS OF RC INT00280
DATA RCU / INT00290
*.274, .274, .229, .1, .5219, .1, .1959, .44767, .14, .2, .941, .2, .435, INT00300
*.2, .106, .2, .0375, .3, .219, .3, .219, .5, .218, .4, .277, .4, .124, .3, .42, .8, .579, INT00310
*.8, .597, .2, .806, .2, .701, .2, .48, .1, .627, .2, .696, .3, .2898 / INT00320
C** VALUE OF C AT LOWER THRESHOLDS. INT00330
DATA CL / INT00340
*.015, .025, .03, .05, .0841, .18, .50, .02, .02, .02, INT00350
*.02, .57, .57, .02, .02, .02, .02, .33, .33, .02, INT00360
*.02, .02, .02, .33, .33 / INT00370
C** VALUE OF C AT UPPER THRESHOLD INT00380
DATA CU / INT00390
*.80, .80, .6324, .70, .80, .80, .80, .100, .100, .100, INT00400
*.100, .100, .16, .88, .100, .100, .100, .100, .100, .16, .17, .100, INT00410
*.100, .100, .100, .100, .100 / INT00420
C** INDEX OF MODEL EQUATION COEFFICIENT INDICES FOR UPPER AMB. ILL. INT00430
DATA IEQ1 /2,2,1,1,4,4,3,3,5,5,0,6,6,0,7,7,8,8,1,2,3,4,4,5,1,2,3, INT00440
*4,4,5,1,2,3,4,4,5/ INT00450
C** INDEX OF MODEL EQUATION COEFFICIENT INDICES FOR LOWER AMB ILL. INT00460
DATA IEQ2 /-1,4,4,3,-1,5,5,0,-1,6,6,-1,7,7,-1,8,-1,9,2,3,4,-1,5, INT00470
*6,2,3,4,-1,5,6,2,3,4,-1,5,6/ INT00480
C** LOWEST RC VALUE IN EACH REGION. INT00490
DATA RLW / INT00500
*.0, .26794, .2, .118, .2, .133, .26794, .4958, .1, .8439, .2, .29, .312, INT00510
*.312, .1, .5119, .0, .0, .1, .1758, .0, .03089, .03089, .05, INT00520
*.3987, .15964, .054, .054, .442, .174, .64, .244, .087, INT00530
*.087, .4394, .1605, .59, .287, .129, .129, .4948, .1894 / INT00540
C** LARGEST RC VALUE IN EACH REGION. INT00550
DATA RUP / INT00560
*.67890, .2, .1335, .2, .1637, .2, .74, .61280, .2, .133, .2, .5764, .2, .74, .739, INT00570
*.2, .29, .84083, .1, .5219, .1, .5219, .712, .1, .1959, .3751, .4477, INT00580
*.2, .942, .2, .435, .2, .106, .9023, .3, .219, .3, .219, .5, .218, .4, .277, .4, .124, INT00590
*.1, .0396, .8, .597, .8, .597, .2, .81, .2, .71, .2, .49, .9575, .2, .696, .3, .29 / INT00600
C** LOWEST VALUE OF C OVER EACH REGION. INT00610
DATA CLW / INT00620
*.015, .025, .250, .270, .025, .030, .270, .6324, .030, INT00630
*.050, .6324, .050, .084107, .70, .0841, .180, .180, .500, INT00640
*.020, .020, .020, .020, .570, .570, .020, .020, .020, INT00650
*.020, .330, .330, .020, .020, .020, .330, .330 / INT00660
C** LARGEST VALUE OF C OVER EACH REGION. INT00670
DATA CUP / INT00680
INT00690
INT00700

```

*.025	.250	.270	.800	.030	.270	.6324	.800	.050	INT00710
*.6324	.700	.084107	.70	.800	.180	.800	.500	.800	INT00720
*100.	.100.	.100.	.570	.100.	.16.88	.100.	.100.	.100.	INT00730
*.330	.100.	.16.17	.100.	.100.	.100.	.330	.100.	.100.	INT00740

```

C** SET LIGHT LEVELS, FRACTIONAL INTERPOLANT.
    ICM1=0
    ICM2=0
    IF (NAL1.EQ.NAL2) GOTO 10
    IF (LSC.GT.5) GOTO 10
    AL1=10.**(3-NAL1)
    AL2=10.**(3-NAL2)
    FACTR=(AL-AL2)/(AL1-AL2)
    DFAC=1.-FACTR
    GOTO (100,100,200,200,200),LSC
10  CALL CINV(LSC,RC,NAL1,C,ICM1,ALFA,BETA,GAMA,AMRAL,AMRGM)
20  NAL2=0
    RETURN
30  CALL CINV(LSC,RC,NAL2,C,ICM2,ALFA,BETA,GAMA,AMRAL,AMRGM)
40  NAL1=0
    RETURN
C** DEVICES 1,2
105 IF (NAL2.LE.7) GOTO 105
    IF (NAL1.EQ.7.AND.FACTR.GT.0.75) GOTO 10
    NAL1=0
    GOTO 20
105 NRG=IRGNS(NAL1)
    IDX=INDEX(NAL1)
    LIM1=NAL1
    LIM2=NAL2
    GOTO 500
C** DEVICES 3,4,5
200 IF (NAL1.GE.4) GOTO 205
    IF (NAL2.EQ.4.AND.FACTR.LT.0.3) GOTO 30
    NAL2=0
    GOTO 40
205 IS=6+(LSC-3)*5+(NAL1-3)
    NRG=IRGNS(IS)
    IDX=INDEX(IS)
    LIM1=7+(LSC-3)*6+(NAL1-3)
    LIM2=LIM1+1
    BVAL=0.
    IDV=LSC-2
C** FIRST, CHECK LIMIT.
500 IF (RC.GE.RCL(LIM1).AND.RC.GE.RCL(LIM2)) GOTO 510
    CALL CASE3(FACTR,CNOT,RCN,RCL(LIM1),RCL(LIM2),CL(LIM1),CL(LIM2))
    IF (RC.GT.RCN) GOTO 520
    CV=CNOT
    ICM1=-1
    ICM2=-1
    GOTO 800
510 IF (RC.LE.RCU(LIM1).AND.RC.LE.RCU(LIM2)) GOTO 520
    CALL CASE3(FACTR,CNOT,RCN,RCU(LIM1),RCU(LIM2),CU(LIM1),CU(LIM2))
    IF (RC.LT.RCN) GOTO 520
    CV=CNOT
    ICM1=1
    ICM2=1
    GOTO 800
C** RC IS IN BOUNDS OF OPERATIONAL LIMITS, BUT NOT NECESSARILLY
C  BETWEEN CURVES. TEST SUB-REGIONS IN TURN.
520 DO 600 I=1,NRG
    J=IDX+I-1
    IF (RC.LT.RLW(J).OR.RC.GT.RUP(J)) GOTO 600
C** RC IS IN SUB-REGION. NOW TEST C
    CV=0.
    CR1=0.
    CR2=0.
    IQ1=IEQ1(J)
    IQ2=IEQ2(J)
    IF (IQ1.GT.0) GOTO 530
    RCN=RCU(LIM1)

```

	CNOT=CU(LIM1)	INT01410
	IF (IQ1.LT.0) RCN=RCL(LIM1)	INT01420
	IF (IQ1.LT.0) CNOT=CL(LIM1)	INT01430
	IF (LSC.LE.2) CALL CASE2(FACTR,RC,CV,IFLG,CP,RCN,CNOT,ALFA(IQ2),	INT01440
	*BETA(IQ2),GAMA(IQ2))	INT01450
	IF (LSC.GT.2)CALL CASE2(FACTR,RC,CV,IFLG,CP,RCN,CNOT,	INT01460
	*AMRAL(IQ2,IDV),BVAL,AMRGM(IQ2,IDV))	INT01470
	IF (IFLG.NE.0) GOTO 600	INT01480
	IF (CV.GT.CUP(J).OR.CV.LT.CLW(J)) GOTO 600	INT01490
	IF (CP.GT.CUP(J).OR.CP.LT.CLW(J)) GOTO 600	INT01500
	IF (RCN.EQ.RCU(LIM1)) ICM1=1	INT01510
	IF (RCN.EQ.RCL(LIM1)) ICM1=-1	INT01520
	GOTO 800	INT01530
530	IF (IQ2.GT.0) GOTO 540	INT01540
	RCN=RCU(LIM2)	INT01550
	CNOT=CU(LIM2)	INT01560
	IF (IQ2.LT.0) RCN=RCL(LIM2)	INT01570
	IF (IQ2.LT.0) CNOT=CL(LIM2)	INT01580
	IF (LSC.LE.2) CALL CASE2(DFAC,RC,CV,IFLG,CP,RCN,CNOT,ALFA(IQ1),	INT01590
	*BETA(IQ1),GAMA(IQ1))	INT01600
	IF (LSC.GT.2)CALL CASE2(DFAC,RC,CV,IFLG,CP,RCN,CNOT,	INT01610
	*AMRAL(IQ1,IDV),BVAL,AMRGM(IQ1,IDV))	INT01620
	IF (IFLG.NE.0) GOTO 600	INT01630
	IF (CV.GT.CUP(J).OR.CV.LT.CLW(J)) GOTO 600	INT01640
	IF (CP.GT.CUP(J).OR.CP.LT.CLW(J)) GOTO 600	INT01650
	IF (RCN.EQ.RCU(LIM2)) ICM2=1	INT01660
	IF (RCN.EQ.RCL(LIM2)) ICM2=-1	INT01670
	GOTO 800	INT01680
540	IF(LSC.LE.2)CALL CASE1(FACTR,RC,CR1,CR2,IFLG,ALFA(IQ1),ALFA(IQ2),	INT01690
	*BETA(IQ1),BETA(IQ2),GAMA(IQ1),GAMA(IQ2))	INT01700
	IF(LSC.GT.2)CALL CASE1(FACTR,RC,CR1,CR2,IFLG,AMRAL(IQ1,IDV),	INT01710
	*AMRAL(IQ2,IDV),BVAL,BVAL,AMRGM(IQ1,IDV),AMRGM(IQ2,IDV))	INT01720
	IF (IFLG.GE.2) GOTO 600	INT01730
	CV=CR1	INT01740
	IF (CV.LE.CUP(J).AND.CV.GE.CLW(J)) GOTO 800	INT01750
	IF (IFLG.EQ.1) GOTO 600	INT01760
	CV=CR2	INT01770
	IF (CV.LE.CUP(J).AND.CV.GE.CLW(J)) GOTO 800	INT01780
600	CONTINUE	INT01790
	C** NO VALUES MET INTERPOLATION CRITERIA.	INT01800
	C** FINAL TEST IS IN THE LIMITING REGIONS	INT01810
640	CONTINUE	INT01820
	WRITE (IOUT,1000) LSC,AL1,AL2	INT01830
1000	FORMAT(1X,40H*** IN ITAM INTAL ROUTINE, DEVICE LSC = ,I2,	INT01840
	*/1X,40H*** COULD NOT INTERPOLATE BETWEEN AMBIENT ILLUM ,F11.6,	INT01850
	*5H AND ,F11.6/1X,39H*** UPPER AMBIENT ILLUM. VALUE ASSUMED.)	INT01860
	GOTO 10	INT01870
800	C=CV	INT01880
	RETURN	INT01890
	END	INT01900

	SUBROUTINE CASE1(FACTR,RC,CR1,CR2,IFLG,ALP1,ALP2,BET1,BET2, *GAM1,GAM2)	CSA00010
C	FINDS CONTRAST C INTERPOLATING ON RC VS C. RETURNS AT MOST TWO	CSA00020
C	ROOTS CR1, CR2. IFLG=1 IF ROOTS ARE IDENTICAL. IFLG=2 IF ROOTS	CSA00030
C	ARE COMPLEX.	CSA00040
C		CSA00050
	DFAC=1.-FACTR	CSA00060
	A1=RC-GAM1*FACTR-GAM2*DFAC	CSA00070
	A2=RC*(ALP1+ALP2)+(BET1-ALP2*GAM1)*FACTR+(BET2-ALP1*GAM2)*DFAC	CSA00080
	A3=ALP1*ALP2*RC+ALP2*BET1*FACTR+ALP1*BET2*DFAC	CSA00090
	IFLG=2	CSA00100
	IF (A1.LT.-1.E-10.OR.A1.GT.1.E-10) GOTO 10	CSA00110
	IF (A2.EQ.0.) RETURN	CSA00120
	IFLG=1	CSA00130
	CR1=-A3/A2	CSA00140
	CR2=CR1	CSA00150
	RETURN	CSA00160
10	DISCR=A2*A2-4.*A1*A3	CSA00170
	IF (DISCR.LT.0.) RETURN	CSA00180
	IF (DISCR.EQ.0.) IFLG=1	CSA00190
	IF (DISCR.GT.0.) IFLG=0	CSA00200
	VA=SQRT(DISCR)/(2.*A1)	CSA00210
	VB=-A2/(2.*A1)	CSA00220
	SG=-1.	CSA00230
	IF (A1.LT.0.) SG=1.	CSA00240
	CR1=VB+SG*VA	CSA00250
	CR2=VB-SG*VA	CSA00260
	RETURN	CSA00270
	END	CSA00280
		CSA00290
		CSA00300

C SUBROUTINE CASE2(FACTR,RC,C,IFLG,CP,RNOT,CNOT,ALP,BET,GAM)
 C FINDS INTERPOLATED CONTRAST C FOR CASE OF SINGLE FIXED RC AT
 C SOME LIMIT.

C IFLG=1
 IF (FACTR.NE.1.) GOTO 10
 IF (RC.NE.RNOT) RETURN
 GOTO 90
 10 IF (RC.EQ.RNOT) GOTO 25
 DIFF=RC-RNOT*FACTR
 IF (DIFF.NE.0.) GOTO 20
 IF (GAM.EQ.0.) RETURN
 CP=BET/GAM
 GOTO 50
 20 VA=BET*(1.-FACTR)+ALP*DIFF
 VB=GAM*(1.-FACTR)-DIFF
 IF (VB.EQ.0.) GOTO 90
 CP=VA/VB
 GOTO 50
 25 IF (GAM.EQ.RC) RETURN
 CP=(ALP*RC+BET)/(GAM-RC)
 GOTO 60
 50 RP=(GAM*CP-BET)/(ALP+CP)
 C=CNOT*(CP-CNOT)/(RC-RNOT)/(RP-RNOT)
 GOTO 100
 60 C=CNOT*FACTR+CP*(1.-FACTR)
 GOTO 100
 90 C=CNOT
 CP=CNOT
 IFLG=0
 100 RETURN
 END

CSB000010
 CSB000020
 CSB000030
 CSB000040
 CSB000050
 CSB000060
 CSB000070
 CSB000080
 CSB000090
 CSB000100
 CSB000110
 CSB000120
 CSB000130
 CSB000140
 CSB000150
 CSB000160
 CSB000170
 CSB000180
 CSB000190
 CSB000200
 CSB000210
 CSB000220
 CSB000230
 CSB000240
 CSB000250
 CSB000260
 CSB000270
 CSB000280
 CSB000290
 CSB000300
 CSB000310
 CSB000320
 CSB000330

	SUBROUTINE CASE3(FACTR,C,RCN,RNOT1,RNOT2,CNOT1,CNOT2)	CSC00010
C		CSC00020
C	FINDS INTERPOLATED C FOR CASE OF BOTH RC VALUES FIXED AT	CSC00030
C	LIMITS.	CSC00040
C		CSC00050
	RCN=RNOT1*FACTR+RNOT2*(1.-FACTR)	CSC00060
	IF (RNOT1.EQ.RNOT2) GOTO 10	CSC00070
	C=CNOT1+(CNOT2-CNOT1)*(RCN-RNOT1)/(RNOT2-RNOT1)	CSC00080
	RETURN	CSC00090
10	C=CNOT1*FACTR+CNOT2*(1.-FACTR)	CSC00100
	RETURN	CSC00110
	END	CSC00120

```

C      SUBROUTINE CYCLE(PB,AJOB,RC)
C      THIS ROUTINE COMPUTES THE NORMALIZED NUMBER OF RESOLVABLE CYCLES
C      FOR GIVEN CUMULATIVE NORMAL DISTRIBUTION OF PROBABILITY OF
C      DETECTION.
C      INPUTS:  PS  = CUMULATIVE PROBABILITY OF DETECTION.
C              PB  = STANDARD DEVIATION FOR PS (.632)
C              AJOB = MEAN RESOLVABLE CYCLES FOR 50 PERCENT PROBABILITY
C                    OF DETECTION.
C      OUTPUTS: RC  = RESOLVABLE CYCLES REQUIRED FOR PS
C      SUBROUTINE: CYCLE CALLED BY ITAM.  CALLS NONE.
C      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLINT,KSTOR,NPLOTU
C      DATA A0,A1,A2 /2.515517,0.802853,0.010328/
C      DATA B1,B2,B3 /1.432788,0.189269,0.001308/
C      ICASE=0
C      IF((0.0,LE,PS).AND.(PS,LE,0.0000003)) GO TO 10
C      IF((.9999997,LE,PS).AND.(PS,LE,1.)) GO TO 20
C      IF((.0000003,LT,PS).AND.(PS,LE,0.5)) GO TO 30
C      IF((.5,LT,PS).AND.(PS,LT,0.9999997)) GO TO 40
C ***** ERROR ROUTINE
C      WRITE(IOOUT,900) PS
C      RETURN
C ***** SPECIAL CASE I
10      X=-5.
C      IF (PS,EQ,0.) X=-45.
C      GO TO 60
C ***** SPECIAL CASE II
20      X=5.
C      IF (PS,EQ,1.) X=45.
C      GO TO 60
C ***** CASE I
30      XX=SQRT(ALOG(1/(PS*PS)))
C      ICASE=1
C      GO TO 50
C ***** CASE II
40      XX=SQRT(ALOG(1./((1.-PS)**2)))
50      X=XX-(A0+A1*XX+A2*(XX*XX))/(1.+B1*XX+B2*(XX*XX)+B3*(XX*XX*XX))
C      IF(ICASE,EQ,1) X=X*(-1.)
60      RC=AJOB+AJOB*PB*X
C      RETURN
900      FORMAT(2X,39HERROR - INCORRECT PS VALUE PASSED  PS = ,F10.4)
C      END

```

```

CYC00010
CYC00020
CYC00030
CYC00040
CYC00050
CYC00060
CYC00070
CYC00080
CYC00090
CYC00100
CYC00110
CYC00120
CYC00130
CYC00140
CYC00150
CYC00160
CYC00170
CYC00180
CYC00190
CYC00200
CYC00210
CYC00220
CYC00230
CYC00240
CYC00250
CYC00260
CYC00270
CYC00280
CYC00290
CYC00300
CYC00310
CYC00320
CYC00330
CYC00340
CYC00350
CYC00360
CYC00370
CYC00380
CYC00390
CYC00400
CYC00410
CYC00420
CYC00430
CYC00440

```



```

SUBROUTINE FPCLOUD(LAMBDA,TRANS,IERR)
REAL LAMBDA,KAPPA,LB0,LP
REAL L,M,N,LO,MO,NO,LOS,MOS,NOS,L0M,L0N,N0M,MXLY,NXLZ,MZNY
DIMENSION IALPH(10)
COMMON /IOUNIT/IOIN,IOOUT,IPHEUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
COMMON /FGEOM/XC,YC,ZC,AE,BE,CE,XR,YR,ZR,XS,YS,ZS
COMMON /OPT/ INDEXP,ETA,KAPPA,W0,THETA0,PHI0,LD,TAUBAR,
+RHO,LB0,TMPA,TMPC
COMMON /GEOMET/PTS(15),IGEOM
DATA IZERO/0/
DATA IALPH/2HCP,2HRP,2HSP,2HAX,2HCL,2HAT,2HBK,
+2HSA,2HLU,2HGO/
ACOS(ARG)=ATAN2(SQRT(1.-ARG**2),ARG)

```

SUBROUTINE FPCLOUD COMPUTES THE FOLLOWING QUANTITIES FOR A FINITE ELLIPSOIDAL CLOUD:

S = LENGTH OF OPTICAL PATH IN CLOUD
TRANS = BEAM TRANSMITTANCE THROUGH CLOUD
LF = PATH RADIANCE
TC = CONTRAST TRANSMITTANCE.

INPUT OF COMPUTATIONAL PARAMETERS TAKES PLACE THROUGH AN ORDER INDEPENDENT READ OF A GROUP OF RECORDS IDENTIFIED BY THE FOLLOWING MNEMONICS (EACH RECORD IS FORMAT (A4,1X,5(E10.4,1X))) :

MNEUMONIC	VARIABLES READ	DESCRIPTION
CPOS	XC,YC,ZC	CLOUD CENTER POSITION
RPOS	XR,YR,ZR	RECEIVER POSITION
SPOS	XS,YS,ZS	SOURCE POSITION
AXES	AE,BE,CE	SEMI-AXES OF CLOUD ELLIPSOID
CLDS	INDEXP,ETA,KAPPA,W0,TMPC	CLOUD AEROSOL PARAMETERS
ATMO	TAUBAR,TMPA	ATMOSPHERIC PARAMETERS
BKGR	RHO,LB0	BACKGROUND PARAMETERS
SANG	THETA0,PHI0	SOLAR ANGLES
LUND	LD	LUNAR DAY
GO		TERMINATES READ

** NOTE : THE GO SENTINEL CARD MUST BE THE LAST CARD READ

THIS INPUT DATA IS STORED FOR LATER USE IN COMMON BLOCKS /FGEOM/ AND /OPT/.

/FGEOM/ INPUT PARAMETERS:

(XC,YC,ZC) = CENTER OF ELLIPSOID
(AE,BE,CE) = SEMI-AXES OF ELLIPSOID
(XR,YR,ZR) = COORDINATES OF RECEIVER LOCATION
(XS,YS,ZS) = COORDINATES OF SOURCE LOCATION.

/OPT/ INPUT PARAMETERS:

INDEXP = PHASE FUNCTION IDENTIFIER

- =0, USER SUPPLIED
- =1, MARITIME ARCTIC, VIS=0.1 TO 2.0 KM
- =2, MARITIME POLAR, VIS=0.2 KM
- =3, MARITIME POLAR, VIS=02. KM
- =4, CONTINENTAL POLAR, VIS= 0.2 TO 2.5 KM
- =5, WHITE PHOSPHORUS
- =6, HEXACHLOROETHANE
- =7, FOG OIL
- =8, DUST (MODERATE AEROSOL LOADING)
- =9, DUST (HEAVY AEROSOL LOADING)
- =10, MARITIME MODEL B, VIS=5KM, RH=95%
- =11, MARITIME MODEL B, VIS=10KM,RH=90%

```

C      =12, MARITIME MODEL B, VIS=50KM,RH=50%
C      ETA = FORWARD SCATTERING PARAMETER
C      ETA MAY ALSO BE CALCULATED BY DEFAULT (I.E. INPUT ETA=0.0).
C      IN THIS CASE INDEXP SHOULD BE THE NEGATIVE OF THE PHASE
C      FUNCTION FOR WHICH ETA IS DESIRED.
C      KAPPA = VOLUME EXTINCTION COEFFICIENT (KM-1)
C      W0 = SINGLE SCATTERING ALBEDO
C      (THETA0,PHI0) = SOLAR ANGLES (DEGREES)
C      LD = LUNAR DAY (INTEGER BETWEEN 0 AND 28)
C      TAUBAR = ATMOSPHERIC OPTICAL THICKNESS ABOVE CLOUD
C      RHO = BACKGROUND SURFACE ALBEDO
C      LBO = BACKGROUND RADIANCE
C      TMPA = TEMPERATURE OF ATMOSPHERE (DEG.C)
C      TMPC = TEMPERATURE OF CLOUD (DEG.C).
C
C      ALL LENGTH UNITS ARE KM. PROGRAM FLOW IS CONTROLLED BY THE
C      VARIABLES TMPA, INDEXP, AND, IMPLICITLY, BY ISW.
C      IF TMPA >= -99.0, THIS SPECIFIES
C      A THERMAL COMPUTATION, WHICH IS PERFORMED IN SUBROUTINE THRMCL;
C      IF TMPA < -99.0 THEN A SCATTERING COMPUTATION IS PERFORMED.
C      IF INDEXP < 0, THIS IS A MULTIPLE SCATTERING COMPUTATION, WHICH
C      IS DONE IN SUBROUTINE MSCLD; IF ETA HAS BEEN INPUT AS ZERO, THEN
C      INDEXP SHOULD BE THE NEGATIVE OF THE PHASE FUNCTION IDENTIFIER,
C      SO THAT ETA WILL BE FOUND FROM THE PROPER PHASE FUNCTION.
C      IF INDEXP > 0 A SINGLE SCATTERING
C      COMPUTATION IS CARRIED OUT IN SUBROUTINE SSCLD. IN THIS CASE
C      INDEXP ALSO SPECIFIES THE PHASE FUNCTION TO BE USED, WITH
C      INDEXP = I SELECTING THE I-TH PHASE FUNCTION. THE VALUE OF
C      ISW OCCURRING IN THE SUBROUTINE PARAMETER LIST INDICATES WHETHER
C      CERTAIN PARAMETERS ARE THE SAME AS IN THE PREVIOUS CALL TO FCLD,
C      AS FOLLOWS.
C
C      ISW IS SET TO 0 WHEN THE FOLLOWING CONDITIONS ARE ENCOUNTERED :
C      A) ALL 9 DATA CARDS ARE READ
C      B) ANY ONE OF CARDS 1-4 AND ANY ONE OF CARDS 5-9
C      (AS LISTED IN THE ORDER ABOVE) ARE READ
C      ISW IS SET TO 2 IF NONE OF CARDS 1-4 IS READ.
C      ISW IS SET TO 1 IF NONE OF CARDS 5-9 IS READ.
C      ISW DEFAULTS TO 2 IF NOTHING IS READ (OTHER THAN THE GO CARD).
C
C      ISW = 2 => PARAMETERS IN COMMON /FGEOM/ ARE THE SAME AS IN
C      PREVIOUS CALL; SKIP PRELIMINARY GEOMETRICAL
C      CALCULATIONS
C      ISW = 1 => PARAMETERS IN COMMON /OPT/ ARE THE SAME AS IN
C      PREVIOUS CALL; SKIP COMPUTATIONS INVOLVING
C      ONLY THESE PARAMETERS
C      ISW = 0 => NEW PARAMETERS IN BOTH /FGEOM/ AND /OPT/;
C      NO CALCULATIONS ARE SKIPPED IN THIS CASE.
C
C      SUBROUTINES CALLED FROM FCLD:
C
C      THRMCL    FOR THERMAL CALCULATION OF PATH RADIANCE
C      MSCLD     FOR MULTIPLE SCATTERING CALCULATION OF
C      PATH RADIANCE
C      SSCLD     FOR SINGLE SCATTERING CALCULATION OF PATH
C      RADIANCE
C      ILLUM     COMPUTES THE EXTRATERRESTRIAL SOLAR OR
C      LUNAR IRRADIANCE. NOT NEEDED FOR THERMAL
C      CALCULATIONS
C      PFN       RETURNS A PHASE FUNCTION VALUE FOR USE BY
C      SSCLD.
C
C      GROUND RADIANCE CAN BE SPECIFIED IN ONE OF TWO WAYS:
C      1.) BY GIVING LBO A NONZERO VALUE, OR
C      2.) AS REFLECTED SOLAR IRRADIANCE, USING THE VALUE OF
C      RHO INPUT TO THE PROGRAM.
C
C      THE VALUE OF LD SPECIFIES WHETHER IT IS DAY OR NIGHT, I. E.
C      WHETHER SOLAR OR LUNAR EXTRATERRESTRIAL IRRADIANCE IS TO BE
C      USED. IF LD = 0 THEN SOLAR IRRADIANCE IS CALCULATED, AND IF

```

```

FCL00700
FCL00710
FCL00720
FCL00730
FCL00740
FCL00750
FCL00760
FCL00770
FCL00780
FCL00790
FCL00800
FCL00810
FCL00820
FCL00830
FCL00840
FCL00850
FCL00860
FCL00870
FCL00880
FCL00890
FCL00900
FCL00910
FCL00920
FCL00930
FCL00940
FCL00950
FCL00960
FCL00970
FCL00980
FCL00990
FCL01000
FCL01010
FCL01020
FCL01030
FCL01040
FCL01050
FCL01060
FCL01070
FCL01080
FCL01090
FCL01100
FCL01110
FCL01120
FCL01130
FCL01140
FCL01150
FCL01160
FCL01170
FCL01180
FCL01190
FCL01200
FCL01210
FCL01220
FCL01230
FCL01240
FCL01250
FCL01260
FCL01270
FCL01280
FCL01290
FCL01300
FCL01310
FCL01320
FCL01330

```

AD-A114 417 ARMY ELECTRONICS RESEARCH AND DEVELOPMENT COMMAND WS--ETC F/G 4/1
PROGRAM LISTINGS FOR EOSAEL 80-B AND ANCILLARY CODES AGAUS AND --ETC(1)
FEB 82 R 0 STEINHOFF
UNCLASSIFIED ERADCOM/ASL-TR-0107-V8-SU NL

UNCLASSIFIED

NL

4 - 6

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{x}} \right) = \frac{\partial L}{\partial x}$$

■


```

C 1 < LD < 28 THE IRRADIANCE VALUE IS FOR LUNAR DAY LD.
C *****
C INITIALIZATION OF INPUT DATA.
C
99 IF (IZERO.NE.0) WRITE (100UT,99)
   FORMAT(1H1)
   IF (IZERO.NE.0) GO TO 477
   XC=0.0
   YC=0.0
   ZC=0.0
   XR=0.0
   YR=0.0
   ZR=0.0
   XS=0.0
   YS=0.0
   ZS=0.0
   AE=0.0
   BE=0.0
   CE=0.0
   INDEXP=0
   ETA=0.0
   KAPPA=0.0
   WD=0.0
   TMPC=0.0
   TAUBAR=0.0
   TMPA=0.0
   RHO=0.0
   LBO=0.0
   THETA0=0.0
   PHI0=0.0
   LD=0
   ISU=0
   IZERO=1
477 CONTINUE
   IFLG=2
   IFLO=1
   DO 360 K=1,10
   READ(10IN,334)IA,IA2,R1,R2,R3,R4,R5
334 FORMAT(2A2,1X,5(E10.4,1X))
   DO 333 I=1,11
   IF (IA.NE.1ALPH(I)) GO TO 333
   IND=I
   IF (IND.EQ.10) GO TO 361
333 CONTINUE
   IF (K.EQ.10.AND.IND.NE.10) GO TO 358
   IF (IND.LT.5) IFLG=0
   IF (IND.GE.5.AND.IND.LE.9) IFLO=0
   IF (IND.EQ.11) GO TO 355
   IF (IND.LT.5) GO TO (341,342,343,344),IND
   INDM4=IND-4
   GO TO (345,346,347,348,349),INDM4
341 XC=R1
   YC=R2
   ZC=R3
   GO TO 360
342 XR=R1
   YR=R2
   ZR=R3
   GO TO 360
343 XS=R1
   YS=R2
   ZS=R3
   GO TO 360
344 AE=R1
   BE=R2
   CE=R3
   GO TO 360
345 INDEXP=IF1X(R1)

```

```

FCL01340
FCL01350
FCL01360
FCL01370
FCL01380
FCL01390
FCL01400
FCL01410
FCL01420
FCL01430
FCL01440
FCL01450
FCL01460
FCL01470
FCL01480
FCL01490
FCL01500
FCL01510
FCL01520
FCL01530
FCL01540
FCL01550
FCL01560
FCL01570
FCL01580
FCL01590
FCL01600
FCL01610
FCL01620
FCL01630
FCL01640
FCL01650
FCL01660
FCL01670
FCL01680
FCL01690
FCL01700
FCL0
FCL0172
FCL01730
FCL01740
FCL01750
FCL01760
FCL01770
FCL01780
FCL01790
FCL01800
FCL01810
FCL01820
FCL01830
FCL01840
FCL01850
FCL01860
FCL01870
FCL01880
FCL01890
FCL01900
FCL01910
FCL01920
FCL01930
FCL01940
FCL01950
FCL01960
FCL01970
FCL01980
FCL01990
FCL02000
FCL02010

```

```

IF (R2.GT..001) ETA=R2
KAPPA=R3
W0=R4
TMPC=R5
GO TO 360
346 TAUBAR=R1
TMPA=R2
GO TO 360
347 RH0=R1
LB0=R2
GO TO 360
348 THETA0=R1
PHI0=R2
GO TO 360
349 LD=IFIX(R1)
GO TO 360
355 WRITE(1000T,357)
357 FORMAT(1H0,20X,46H***FCL00D ERROR*** INPUT RECORD DETECTED WHICH,
+ 44HDOES NOT CORRESPOND TO CORRECT INPUT FORMAT,/)
GO TO 360
358 WRITE(1000T,359)
359 FORMAT(1H0,35X,45H***FCL00D ERROR*** TOO MANY INPUT CARDS OR GO,
+ 16H SENTINEL ABSENT,/)
IERR=1
GO TO 200
360 CONTINUE
361 CONTINUE
ISW=IFLG+IFLO
IF (ISW.EQ.3) ISW=2
IF (IGEOSW.NE.1) GO TO 222
XC=PTS(13)
YC=PTS(14)
ZC=PTS(15)
XR=PTS(4)
YR=PTS(5)
ZR=PTS(6)
XS=PTS(1)
YS=PTS(2)
ZS=PTS(3)
222 CONTINUE
C
C ECHO INPUT
C
WRITE(1000T,1000) XC,YC,ZC,AE,BE,CE,XR,YR,ZR,XS,YS,ZS
IF (ETA.LT.1.E-20) CALL PFNN(LAMBDA,0.,INDEXP,PFN,ETA)
WRITE(1000T,1100) INDEXP,ETA,LAMBDA,KAPPA,W0,TAUBAR,THETA0,PHI0,
1RH0,LB0,TMPA,TMPC,LD
IF (ISW.EQ.2) GO TO 15
C
C ISW .NE. 2 INDICATES PRELIMINARY GEOMETRICAL CALCULATIONS TO
C BE PERFORMED: COMPUTE INTERSECTIONS (XM,YM,ZM) AND (XN,YN,ZN) OF
C LINE OF SIGHT WITH CLOUD, SBAR = LENGTH OF PATH FROM SOURCE TO
C RECEIVER, S = LENGTH OF PATH IN CLOUD, AND TRANS = TRANSMITTANCE
C THROUGH CLOUD
C
L=XS-XR
M=YS-YR
N=ZS-ZR
SBAR=SQRT(L*L+M*M+N*N)
L=L/SBAR
M=M/SBAR
N=N/SBAR
DX=XS-XC
DY=YS-YC
DZ=ZS-ZC
ASQ=AE*AE
BSQ=BE*BE
CSQ=CE*CE
ABSQ=ASQ+BSQ
ACSQ=ASQ+CSQ

```

```

FCL02020
FCL02030
FCL02040
FCL02050
FCL02060
FCL02070
FCL02080
FCL02090
FCL02100
FCL02110
FCL02120
FCL02130
FCL02140
FCL02150
FCL02160
FCL02170
FCL02180
FCL02190
FCL02200
FCL02210
FCL02220
FCL02230
FCL02240
FCL02250
FCL02260
FCL02270
FCL02280
FCL02290
FCL02300
FCL02310
FCL02320
FCL02330
FCL02340
FCL02350
FCL02360
FCL02370
FCL02380
FCL02390
FCL02400
FCL02410
FCL02420
FCL02430
FCL02440
FCL02450
FCL02460
FCL02470
FCL02480
FCL02490
FCL02500
FCL02510
FCL02520
FCL02530
FCL02540
FCL02550
FCL02560
FCL02570
FCL02580
FCL02590
FCL02600
FCL02610
FCL02620
FCL02630
FCL02640
FCL02650
FCL02660
FCL02670
FCL02680
FCL02690
FCL02700

```



```

40 THETA= ACOS(N)
   PHI=0.0
   IF(M.NE.0.0.OR.L.NE.0.0) PHI=ATAN2(M,L)
   COSCHI=N*N0+SIN(THETA)*SIN(TH0)*COS(PHI-PH0)
   CHI= ACOS(COSCHI)*57.2958
   CALL PFNN(LAMBDA,CHI,INDEXP,PFN,ETA)
   P=PFN
   RGRND=E0*TATM*N0
   IF(LB0.LT.1.E-10) LB0=RHO*RGRND
   DX=XM-XC
   DY=YM-YC
   DZ=ZM-ZC
   LOS=L0*L0
   MOS=M0*M0
   NOS=N0*N0
   LOM=L0*M-M0*L
   LON=L0*N-N0*L
   NOM=N0*M-M0*N
   MXLY=M0*DX-L0*DY
   NXLZ=N0*DX-L0*DZ
   MZNY=M0*DZ-N0*DY
   DENOM=BCSQ*LOS+ACSQ*MOS+ABSQ*NOS
   ALPHA=(BCSQ*L0*L0+ACSQ*M0*M0+ABSQ*N0*N0)/DENOM
   BETA=-(BCSQ*L0*DX+ACSQ*M0*DY+ABSQ*N0*DZ)/DENOM
   GAMMA=-ABCSQ*(CSQ*L0M*L0M+BSQ*L0N*L0N+ASQ*N0M*N0M)/(DENOM*DENOM)
   DELTA=CSQ*MXLY*L0M+BSQ*NXLZ*L0N+ASQ*MZNY*N0M
   DELTA=-2.0*ABCSQ*DELTA/(DENOM*DENOM)
   EPS=BCSQ*LOS+ACSQ*MOS+ABSQ*NOS-CSQ*MXLY*MXLY-BSQ*NXLZ*NXLZ
   EPS=EPS/(DENOM*DENOM)
   IF(EPS.GE.0.0) GO TO 50
   WRITE(1000,1300) EPS
   EPS=0.0
50 IF(GAMMA.LT.0.0) GO TO 60
   WRITE(1000,1400) GAMMA
   GO TO 200
60 CALL SSCLD(ALPHA,BETA,GAMMA,DELTA,EPS,S,KAPPA,W0,RGRND,P,LP)
C
C
C
WRITE RESULTS
90 F=LP/(LB0*TRANS)
   TC=1.0/(1.0+F)
   WRITE(1000,1500) S,TRANS,LP,TC
200 RETURN
1000 FORMAT(1H0,43X,45H-- RADIATIVE TRANSFER THROUGH FINITE CLOUD --/
1 1H0,45X,14H(XC,YC,ZC) = (,2(F8.4,1H),F8.4,12H) KILOMETERS/
2 45X,14H(AE,BE,CE) = (,2(F8.4,1H),F8.4,1H)/
3 45X,14H(XR,YR,ZR) = (,2(F8.4,1H),F8.4,1H)/
4 45X,14H(XS,YS,ZS) = (,2(F8.4,1H),F8.4,1H)/
1100 FORMAT(1H0,45X,8HINDEXP =,I9,10X,8HETA =,F11.3/
1 45X,8HLAMBDA =,F9.3,8H(MU) ,2X,9HKAPPA =,1PE10.4,7H(KM-1)/
2 45X,8HOMEGA 0=,0PF9.3,10X,8HTAUBAR =,1X,1PE10.4/
3 45X,8HETHETA0 =,0PF9.1,10X,8HPHI0 =,F11.1,10H(DEGREES)/
4 45X,8HRHO =,0PF9.3,10X,8HLB0 =,F11.3,13H(W/M2-SR-MU)/
5 45X,8HTMPA =,F9.1,8H(DEG.C),2X,8HTMPC =,F11.1/
6 45X,8HLD =,I9)
1200 FORMAT(1H0,44X,43H**LINE-OF-SIGHT MISSES CLOUD. S SET TO 0.0)
1300 FORMAT(1H0,44X,4HEPS=,E10.4,24H LT 0.0. EPS SET TO 0.0)
1400 FORMAT(1H0,44X,6HGAMMA=,E10.4,29H GE 0.0. SKIP TO NEXT CASE.)
1500 FORMAT(1H0,1H0,37X,11HPATH LENGTH,3X,15HTRANSMITTANCE ,
1 13HPATH RADIANCE,4X,8HCONTRAST/
2 38X,11H(IN CLOUD) ,19X,12H(W/M2-SR-MU),15H TRANSMITTANCE/
3 36X,4(15H-----+))
4 1H0,40X,F6.3,7X,1PE9.3,5X,1PE9.3,5X,1PE9.3)
1600 FORMAT(1H0,45X,28HUNIT SOURCE VECTOR L, M, N =,3(1X,F7.4)/
1 1H0,45X,23HSOLAR VECTOR L0,M0,N0 =,3(1X,F7.4))
END

```

FCL03410
FCL03420
FCL03430
FCL03440
FCL03450
FCL03460
FCL03470
FCL03480
FCL03490
FCL03500
FCL03510
FCL03520
FCL03530
FCL03540
FCL03550
FCL03560
FCL03570
FCL03580
FCL03590
FCL03600
FCL03610
FCL03620
FCL03630
FCL03640
FCL03650
FCL03660
FCL03670
FCL03680
FCL03690
FCL03700
FCL03710
FCL03720
FCL03730
FCL03740
FCL03750
FCL03760
FCL03770
FCL03780
FCL03790
FCL03800
FCL03810
FCL03820
FCL03830
FCL03840
FCL03850
FCL03860
FCL03870
FCL03880
FCL03890
FCL03900
FCL03910
FCL03920
FCL03930
FCL03940
FCL03950
FCL03960
FCL03970
FCL03980
FCL03990
FCL04000
FCL04010
FCL04020
FCL04030
FCL04040
FCL04050
FCL04060
FCL04070

```

SUBROUTINE THRMCL(RBAR,W0,TMPA,TMPC,LAMBDA,KAPPA,TRANS,LP)      THR00010
REAL LAMBDA,LP,LI,KAPPA                                         THR00020
COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU THR00030
BB(X,T)=1.19106E8/(X**5*(EXP(1.4388E4/(X*T))-1.0))              THR00040
C*****                                                         THR00050
C SUBROUTINE PERFORMS THERMAL RADIATION CALCULATIONS FOR FINITE  THR00060
C CLOUD, RETURNING THE VALUE OF LP = PATH RADIANCE.             THR00070
C INPUTS ARE:                                                    THR00080
C C                                                                THR00090
C C RBAR = AVERAGE HALF-LENGTH OF PATH THROUGH CLOUD           THR00100
C C W0 = SINGLE SCATTERING ALBEDO                                THR00110
C C TMPA = TEMPERATURE OF THE ATMOSPHERE                         THR00120
C C TMPC = TEMPERATURE OF THE CLOUD                             THR00130
C C LAMBDA = WAVELENGTH                                           THR00140
C C KAPPA = VOLUME EXTINCTION COEFFICIENT                       THR00150
C C TRANS = TRANSMITTANCE THROUGH CLOUD.                         THR00160
C C                                                                THR00170
C C*****                                                         THR00180
C G=1.0-EXP(-KAPPA*RBAR)                                         THR00190
B=BB(LAMBDA,273.16+TMPC)                                         THR00200
LI=BB(LAMBDA,273.16+TMPA)                                         THR00210
WRITE(IOOUT,1000) B,LI                                           THR00220
LP=(1.0-TRANS)*((1.0-W0)*(1.0+G*W0)*B+W0*(1.0-G)*LI)           THR00230
1000 FORMAT(1H0,43,40H **THERMAL CALCULATION OF PATH RADIANCE/,45X, THR00240
122H BB(LAMBDA,TMPC) = ,1PE10.4,11H W/M2-SR-MU/,45X,             THR00250
122H BB(LAMBDA,TMPA) = ,1PE10.4,11H W/M2-SR-MU)                 THR00260
RETURN                                                            THR00270
END                                                                THR00280

```

```

SUBROUTINE SSCLD(ALPHA,BETA,GAMMA,DELTA,EPS,S,KAPPA,
1W0,RGRND,P,LP)
COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
C*****
C SUBROUTINE COMPUTES PATH RADIANCE LP FOR THE CASE OF SINGLE
C SCATTERING IN A FINITE CLOUD.  INPUTS ARE GEOMETRICAL PARAMETERS
C ALPHA, BETA, GAMMA, DELTA, AND EPS, S = LENGTH OF OPTICAL PATH
C THROUGH CLOUD, AND:
C KAPPA = VOLUME EXTINCTION COEFFICIENT
C W0 = SINGLE SCATTERING ALBEDO
C RGRND = SURFACE BACKGROUND RADIANCE
C P = SINGLE SCATTERING PHASE FUNCTION.
C*****
REAL KAPPA,LP
COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK
ASIN(ARG)=ATAN2(ARG,SQRT(1.-ARG**2))
WRITE(IOOUT,1500)
DISCRM=DELTA*DELTA-4.0*GAMMA*EPS
IF(DISCRT.GT.0.0) GO TO 10
WRITE(IOOUT,1000) DISCRM
LP=0.0
GO TO 20
10 TGSDEL=2.0*GAMMA*S+DELTA
HS= ASIN(TGSDEL/SQRT(DISCRT))/SQRT(-GAMMA)
H0= ASIN(DELTA/SQRT(DISCRT))/SQRT(-GAMMA)
TRANS=EXP(-KAPPA*S)
BETAB=W0*KAPPA
LP=(1.0-BETA*KAPPA)*S+(1.0-ALPHA)*KAPPA*S*S/2.0
LP=LP-KAPPA*DISCRM*(HS-H0)/(8.0*GAMMA)
LP=LP-KAPPA*(TGSDEL*SQRT(GAMMA*S*S+DELTA*S+EPS)-
1 DELTA*SQRT(EPS))/(4.0*GAMMA)
LP=BETAB*LP*P*RGRND*TRANS/(4.0*PI)
20 RETURN
1000 FORMAT(9H0DISCRM =,E10.4,25H IN SSCLD. LP SET TO 0.0)
1500 FORMAT(1H0,43X,33H **RESULTS FOR SINGLE SCATTERING)
END

```

```

SSC00010
SSC00020
SSC00030
SSC00040
SSC00050
SSC00060
SSC00070
SSC00080
SSC00090
SSC00100
SSC00110
SSC00120
SSC00130
SSC00140
SSC00150
SSC00160
SSC00170
SSC00180
SSC00190
SSC00200
SSC00210
SSC00220
SSC00230
SSC00240
SSC00250
SSC00260
SSC00270
SSC00280
SSC00290
SSC00300
SSC00310
SSC00320
SSC00330
SSC00340
SSC00350
SSC00360
SSC00370

```

```

SUBROUTINE PFNN(LAMBDA,CHI,INDEXP,PFN,ETA)
REAL LAMBDA,LAMDA1,LAM1,LAM2,KAPPA
COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUP
COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK
DIMENSION PHANG(65),PF1(65),PF2(65)
DATA LAMDA1,IP1,CHI1/-1.0,-2,-1.0/
C*****
C SUBROUTINE COMPUTES THE VALUE OF THE INDEXP-TH PHASE FUNCTION
C AT SCATTERING ANGLE CHI AND WAVELENGTH LAMBDA USING BILINEAR
C INTERPOLATION.
C*****
      ACOS(ARG)=ATAN2(SQRT(1.-ARG**2),ARG)
      IERR=0
      MAXID=12
      PFNDAT ONLY CONTAINS ONE PFN FOR THE VISIBLE (.55UM) AND
      THE NEAR IR (1.06UM); THEREFORE DO NOT INTERPOLATE.
      IWAVE=0
      IF (LAMBDA.LT.2.0) IWAVE=1
      PRELIMINARIES FOR FINDING ETA
      ETA1=0.0
      ETA2=0.0
      IDCK=0
      IF (INDEXP.LT.0) IDCK=1
      IF (INDEXP.LT.0) INDEXP=-INDEXP
      IF (INDEXP.EQ.IP1.AND.CHI.EQ.CHI1.AND.LAMBDA.EQ.LAMDA1)
      GO TO 200
      IF (INDEXP.EQ.IP1.AND.LAMBDA.EQ.LAMDA1) GO TO 70
      REWIND IPHFUN
      NRD=0
      DO 20 I=1,66,11
      READ(IPHFUN,1000) (PHANG(I+J-1),J=1,11)
      DO 10 J=1,11
      NRD=NRD+1
      IF (PHANG(NRD).GE.999.9) GO TO 30
      10 CONTINUE
      20 CONTINUE
      30 NA=NRD-1
      DO 35 I=1,NA
      PHANG(I)=COS(PHANG(I)*PIRAD)
      35 CONTINUE
      END-OF-FILE CHECK
      IF (IERR.EQ.2) GO TO 195
      READ(IPHFUN,1100) IANG1,ID,LAM1,W0,KAPPA,BETA
      IF (LAM1.GE.12.00.AND.ID.EQ.MAXID) IERR=2
      READ(IPHFUN,1200) (PF1(I),I=1,NA)
      SUM=0.
      START RENORMALIZATION OF PHASE FUNCTION - ALSO SEE BELOW
      DO 45 J=2,NA
      SUM=SUM+(-PHANG(J)+PHANG(J-1))*(PF1(J)+PF1(J-1))/4.
      45 DO 46 J=1,NA
      PF1(J)=PF1(J)/SUM
      ETA1=ETAINT(PF1,PHANG,NA)
      IF (ID.NE.INDEXP) GO TO 40
      IF (IWAVE.EQ.1.AND.LAMBDA.GT.LAM1) GO TO 40
      IF (IWAVE.EQ.1) GO TO 75
      IF (LAMBDA.LT.LAM1) GO TO 190
      50 CONTINUE
      IF (IERR.EQ.2) GO TO 195
      READ(IPHFUN,1100) IANG2,ID,LAM2,W0,KAPPA,BETA
      IF (LAM2.GE.12.00.AND.ID.EQ.MAXID) IERR=2
      READ(IPHFUN,1200) (PF2(I),I=1,NA)
      SUM=0.
      DO 55 J=2,NA
      SUM=SUM+(-PHANG(J)+PHANG(J-1))*(PF2(J)+PF2(J-1))/4.
      55 DO 56 J=1,NA
      PF2(J)=PF2(J)/SUM
      56 ETA2=ETAINT(PF2,PHANG,NA)
      C THE PHASE FUNCTION(S) ARE NOW NORMALIZED TO: INTEGRAL OF
      PFN00010
      PFN00020
      PFN00030
      PFN00040
      PFN00050
      PFN00060
      PFN00070
      PFN00080
      PFN00090
      PFN00100
      PFN00110
      PFN00120
      PFN00130
      PFN00140
      PFN00150
      PFN00160
      PFN00170
      PFN00180
      PFN00190
      PFN00200
      PFN00210
      PFN00220
      PFN00230
      PFN00240
      PFN00250
      PFN00260
      PFN00270
      PFN00280
      PFN00290
      PFN00300
      PFN00310
      PFN00320
      PFN00330
      PFN00340
      PFN00350
      PFN00360
      PFN00370
      PFN00380
      PFN00390
      PFN00400
      PFN00410
      PFN00420
      PFN00430
      PFN00440
      PFN00450
      PFN00460
      PFN00470
      PFN00480
      PFN00490
      PFN00500
      PFN00510
      PFN00520
      PFN00530
      PFN00540

```

C	PHASE FUNCTION OVER ALL SOLID ANGLE DIVIDED BY 4 PI = 1.	PFN00550
	IF (ID.NE.INDEXP) GO TO 190	PFN00560
	IF (LAMBDA.LE.LAM2) GO TO 70	PFN00570
	LAM1=LAM2	PFN00580
	IANG1=IANG2	PFN00590
	DO 60 I=1,NA	PFN00600
	PF1(I)=PF2(I)	PFN00610
60	CONTINUE	PFN00620
	GO TO 50	PFN00630
70	CONTINUE	
	IF (IWAVE.EQ.1) GO TO 75	
	DLAM=(LAMBDA-LAM1)/(LAM2-LAM1)	PFN00640
75	CONTINUE	
	IF (CHI.LT.-1.E-3.OR.CHI.GT.180.001) GO TO 190	PFN00650
	DO 80 J=2,NA	PFN00670
	IF (CHI.LE.(ACOS(PHANG(J))/PIRAD)) GO TO 90	PFN00680
80	CONTINUE	PFN00690
	J=NA	PFN00700
90	J1=J-1	PFN00710
	DCHI=(COS(CHI*PIRAD)-PHANG(J1))/(PHANG(J)-PHANG(J1))	PFN00720
	IF (IWAVE.NE.1) GO TO 95	
	PFN=PF1(J1)+DCHI*(PF1(J)-PF1(J1))	
	GO TO 96	
95	PFN=PF1(J1)+DLAM*(PF2(J1)-PF1(J1))+DCHI*(PF1(J)-PF1(J1))	PFN00730
	1+DLAM*DCHI*(PF2(J)+PF1(J1)-PF2(J1)-PF1(J))	PFN00740
96	LAMDA1=LAMBDA	PFN00760
	CHI1=CHI	PFN00770
	IF (IP1.NE.-2) WRITE(IOOUT,1500) CHI,PFN	
	IP1=INDEXP	
	ETA=ETA1+DLAM*(ETA2-ETA1)	
	GO TO 200	PFN00790
190	WRITE(IOOUT,1300) ID,INDEXP,LAM1,LAMBDA,CHI	PFN00800
	STOP	PFN00810
195	WRITE (IOOUT,1600) IPHFUN	PFN00820
	STOP	PFN00830
200	CONTINUE	
	IF (IDCK.EQ.1) INDEXP=-INDEXP	
	RETURN	PFN00840
1000	FORMAT(11(F6.2,1X))	PFN00850
1100	FORMAT(2(I2,1X),F5.2,1X,F8.6,1X,2(E12.6,1X))	PFN00860
1200	FORMAT(6(E12.6,1X))	PFN00870
1300	FORMAT(33H0ERROR IN READING PHASE FUNCTION./	PFN00880
	127H ID,INDEXP,LAM1,LAMBDA,CHI=,2I3,3E13.7)	PFN00890
1500	FORMAT(1H0,23HSCATTERING ANGLE CHI = ,F8.2,7X,24H PHASE FN P(LAMBDA	PFN00900
	+A,CHI)=,E10.4)	PFN00910
1600	FORMAT(1X,32HATTEMPT TO READ PAST EOF ON UNIT,13,18H IN SUBROUTINE	PFN00920
	1 PFN//)	PFN00930
	END	PFN00940


```

SUBROUTINE MSCLD(TAU,TAU0,TRANS,TATM,E0,W0,ETA,RHO,LP) MSC00010
REAL LP,LBAR,K MSC00020
COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK MSC00030
COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUMSC00040
SINH(ARG)=.5*(EXP(ARG)-EXP(-ARG))
COSH(ARG)=.5*(EXP(ARG)+EXP(-ARG))
C***** MSC00050
SUBROUTINE COMPUTES LP = PATH RADIANCE DUE TO MULTIPLE SCATTERING MSC00060
IN A FINITE CLOUD. INPUTS ARE: MSC00070
MSC00080
TAU = CLOUD OPTICAL DEPTH MSC00100
TAU0 = CLOUD OPTICAL THICKNESS MSC00110
TRANS = TRANSMITTANCE ALONG LINE OF SIGHT MSC00120
TATM = TRANSMITTANCE OF ATMOSPHERE ABOVE CLOUD MSC00130
E0 = EXTRATERRESTRIAL IRRADIANCE MSC00140
W0 = SINGLE SCATTERING ALBEDO MSC00150
ETA = FORWARD SCATTERING PARAMETER MSC00160
RHO = BACKGROUND SURFACE REFLECTANCE. MSC00170
MSC00180
C***** MSC00190
IF(W0,LT,1.0) GO TO 10 MSC00200
W0=.999 MSC00210
WRITE(IOOUT,2000) MSC00220
10 C1=1.0-ETA*W0 MSC00230
C2=(1.0-ETA)*W0 MSC00240
K=SQRT((1.0-W0)*(1.0+W0-2.0*ETA*W0)) MSC00250
TOK=K*TAU0 MSC00260
TOTK=K*(TAU0-TAU) MSC00270
TK2=2.0*K*TAU MSC00280
GAMMA=E0*TATM/(C1*SINH(TOK)+K*COSH(TOK)) MSC00290
DELTA=RHO*K/((C1-RHO*C2)*SINH(2.0*TOK)+K*COSH(2.0*TOK)) MSC00300
WRITE(IOOUT,1000) MSC00310
EPLUS=C2*SINH(TOK)+DELTA*(C1*SINH(TK2)+K*COSH(TK2)) MSC00320
EMINUS=C1*SINH(TOK)+K*COSH(TOK)+DELTA*C2*SINH(TK2) MSC00330
LBAR=GAMMA*(EPLUS+EMINUS)/TWOPI MSC00340
LP=W0*LBAR*(1.0-TRANS) MSC00350
WRITE(IOOUT,100)TAU,TAU0,TATM MSC00360
100 FORMAT(1H0,45X,18H OPTICAL DEPTH = ,1PE10.4/,45X, MSC00370
131H OPTICAL THICKNESS OF CLOUD = ,1PE10.4/,45X, MSC00380
244H TRANSMITTANCE OF ATMOSPHERE ABOVE CLOUD = ,1PE10.4) MSC00390
RETURN MSC00400
1000 FORMAT(36H0 **RESULTS FOR MULTIPLE SCATTERING ) MSC00410
2000 FORMAT(47H0 **OMEGA 0 WAS 1.0, NOW SET TO 0.999 IN MSCLD) MSC00420
END MSC00430

```

```

C      FUNCTION ETAINT(PFN,PHANG,NA)
C      THIS FUNCTION WILL DETERMINE ETA, THE FORWARD SCATTERING
C      PARAMETER:  $\text{ETA} = .5 \times \text{INTEGRAL PFN OVER THETA, WHERE THETA GOES}$ 
C      FROM ZERO TO  $\text{PI}/2$ .
C      COMMON /CONST/ PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK
C      DIMENSION PHANG(65),PFN(65)
C      NAM1=NA-1
C      ETA=0.
C      DO 1 I=1,NAM1
C      IF(PHANG(I+1).GT.0.) ETA=ETA+(PHANG(I)-PHANG(I+1))*
1      (PFN(I+1)+PFN(I))/4.
C      CONTINUE
C      ETAINT=ETA
C      RETURN
C      END

```

```

SUBROUTINE OVR0ST(LAMBDA,TRANS,IERR)
REAL LC,LG,LB0,KAPPA,LAMBDA,MU,LP
DIMENSION IALPH(7)
COMMON /IOUNIT/IOIN,IOOUT,IPHFN,LOUNIT,NDIRTU,NCLINT,KSTOR,NPLOTU
COMMON /GEOMET/PTS(15),IGEOSW
DATA IZERO/0/
DATA IALPH/2HOP,2HSP,2HCL,2HBK,2HGR,2HTE,2HGD /
BB(X,T)=1.191062E8/((X**5*(EXP(1.4387864E4/(X*T))-1.0)))
*****
SUBROUTINE COMPUTES BEAM TRANSMITTANCE, PATH RADIANCE, AND
CONTRAST TRANSMITTANCE ALONG AN OPTICAL PATH UNDER AN OVERCAST
SKY. ORDER-INDEPENDENT INPUT CARDS ARE AS FOLLOWS:
(INDIVIDUAL RECORD FORMAT IS (A4,1X,5(E10.4,1X)))
MNEMONIC      VARIABLES READ      DESCRIPTION
-----
OPOS           XO,YO,ZO            OBSERVER POSITION
SPOS           XT,YT,ZT            SOURCE POSITION
CLDS           ZC,LC,KAPPA,ETA ,WO  CLOUD PARAMETERS
BKGR           LB0                 BACKGROUND RADIANCE
GRND           LG                  GROUND RADIANCE
TEMP           TEMP                TEMPERATURE ALONG PATH
GO             GO                  END OF READ SENTINEL

THE VARIABLES ZC AND LC REFER TO THE OVERCAST SKY LAYER!
THE VARIABLES KAPPA, ETA, WO, TEMP, REFER TO THE ATMOSPHERE
BETWEEN THE OVERCAST SKY AND GROUND I.E. THE INTERVENING
ATMOSPHERIC PROPERTIES (GAS OR AEROSOL)
** NOTE : THE GO CARD MUST BE THE LAST RECORD READ.

THE FOLLOWING ENUMERATES THE VARIABLES LISTED ON THE ABOVE CARDS :
(XO,YO,ZO) = OBSERVER COORDINATES
(XT,YT,ZT) = SOURCE COORDINATES
ZC = HEIGHT OF CLOUD LAYER
LC = CLOUD RADIANCE
LG = GROUND RADIANCE
LB0 = BACKGROUND RADIANCE
KAPPA = VOLUME EXTINCTION COEFFICIENT (KM-1)
ETA = FORWARD SCATTERING PARAMETER
WO = SINGLE SCATTERING ALBEDO
TEMP = TEMPERATURE ALONG PATH (DEG. C)

LENGTH UNITS ARE KILOMETERS; RADIANCE UNITS ARE W/M2-SR-MU.
IF TEMP >= -99., THERMAL RADIANCE IS CALCULATED; IF TEMP < -99.
SINGLY SCATTERED RADIANCE IS CALCULATED.

SUBROUTINE RETURNS:
TRANS = BEAM TRANSMITTANCE
TO CALLING PROGRAM.
*****
DATA INITIALIZATION
IFCIZERO.NE.0) GO TO 477
XO=0.0
YO=0.0
ZO=0.0
XT=0.0
YT=0.0
ZT=0.0
ZC=0.0
LC=0.0
KAPPA=0.0
ETA=0.0
WO=0.0
LB0=0.0

```

LG=0.0	OVR00670
TEMP=0.0	OVR00680
IZERO=1	OVR00690
477 CONTINUE	OVR00700
DO 360 K=1,7	OVR00710
READ(10IN,334)IA,IA2,R1,R2,R3,R4,R5	OVR0
334 FORMAT(2A2,1X,5(E10.4,1X))	OVR0073
DO 333 I=1,8	OVR00740
IF(IA.NE.IALPH(I)) GO TO 333	OVR00750
IND=I	OVR00760
IF(IND.EQ.7) GO TO 361	OVR00770
333 CONTINUE	OVR00780
IF(IND.EQ.8) GO TO 355	OVR00790
IF(K.EQ.7.AND.IND.NE.7) GO TO 358	OVR00800
GO TO (341,342,343,344,345,346),IND	OVR00810
341 XO=R1	OVR00820
YO=R2	OVR00830
ZO=R3	OVR00840
GO TO 360	OVR00850
342 XT=R1	OVR00860
YT=R2	OVR00870
ZT=R3	OVR00880
GO TO 360	OVR00890
343 ZO=R1	OVR00900
LC=R2	OVR00910
KAPPA=R3	OVR00920
ETA=R4	OVR00930
WO=R5	OVR00940
GO TO 360	OVR00950
344 LB0=R1	OVR00960
GO TO 360	OVR00970
345 LG=R1	OVR00980
GO TO 360	OVR00990
346 TEMP=R1	OVR01000
GO TO 360	OVR01010
355 WRITE(IOOUT,357)	OVR01020
357 FORMAT(1H0,25X,44H***OVR00670 ERROR*** INPUT CARD DETECTED WHICH,	OVR01030
+ 36H DOES NOT MATCH CORRECT INPUT FORMAT,/) GO TO 360	OVR01040
358 WRITE(IOOUT,359)	OVR01050
359 FORMAT(1H0,34X,45H***OVR00670 ERROR*** TOO MANY INPUT CARDS OR GO,	OVR01060
+ 16H SENTINEL ABSENT,/) IERR=1	OVR01070
GO TO 200	OVR01080
360 CONTINUE	OVR01090
361 CONTINUE	OVR01100
IF(IGEOSM.NE.1) GO TO 222	OVR01110
XO=PTS(4)	OVR01120
YO=PTS(5)	OVR01130
ZO=PTS(6)	OVR01140
XT=PTS(1)	OVR01150
YT=PTS(2)	OVR01160
ZT=PTS(3)	OVR01170
222 CONTINUE	OVR01180
	OVR01190
	OVR01200
	OVR01210
	OVR01220
	OVR01230
	OVR01240
	OVR01250
	OVR01260
	OVR01270
	OVR01280
	OVR01290
	OVR01300
	OVR01310
	OVR01320
	OVR01330
	OVR01340
	OVR01350
	OVR01360

C
C
C
ECHO INPUT
WRITE(IOOUT,1000) XO,XT,YO,YT,ZO,ZT,ZC,LC,LAMBDA,LG,
1 TEMP,LB0,KAPPA,WO,ETA
IF(TEMP.LT.-99.0) GO TO 4
BBTEMP=BB(LAMBDA,273.16+TEMP)
WRITE(IOOUT,1800) BBTEMP
BTE=(WO-1.0)*BBTEMP
GO TO 8
4 WRITE(IOOUT,1700)
8 F=2.0*(1.0-ETA)
ZLEN=ZT-ZO
S=SQRT((XT-XO)**2+(YT-YO)**2+ZLEN**2)
MU=ABS(ZLEN)/S
TO=KAPPA*ZO

1 1H0,50X,8HBBTEMP= ,1PE10.4,11H W/M2-SR-MU>
END

OVR02070
OVR02080

FUNCTION G2(TAU1,TAU2,MU)	FUG00010
*****	FUG00020
SUBROUTINE COMPUTES IN CLOSED FORM INTEGRALS OF THE FUNCTION	FUG00030
EXP(TAU/MU)*E2(TAU)	FUG00040
	FUG00050
	FUG00060
	FUG00070
WHERE E2 IS THE SECOND EXPONENTIAL INTEGRAL, FOR DETAILS SEE	FUG00080
KOURGANOFF, 'BASIC METHODS IN TRANSFER PROBLEMS', APPENDIX I	FUG00090
(PAGES 256-257 OF FIRST EDITION, 1952, OXFORD UNIVERSITY PRESS.)	FUG00100
*****	FUG00110
REAL MU	FUG00120
DATA GAMMA/.5772156649/	FUG00130
IF(MU.LT..9999) GO TO 50	FUG00140
	FUG00150
MU EQ 1.0	FUG00160
	FUG00170
IF(TAU1.NE.0.0) GO TO 10	FUG00180
G2=(TAU2-1.0)*EXP(TAU2)*E1(TAU2)-GAMMA-ALOG(TAU2)	FUG00190
GO TO 100	FUG00200
10 IF(TAU2.NE.0.0) GO TO 20	FUG00210
G2=GAMMA+ALOG(TAU1)+EXP(TAU1)*(1.0-TAU1)*E1(TAU1)	FUG00220
GO TO 100	FUG00230
20 G2=EXP(TAU1)*(1.0-TAU1)*E1(TAU1)-EXP(TAU2)*(1.0-TAU2)*E1(TAU2)	FUG00240
G2=G2+(ALOG(TAU1)-ALOG(TAU2))	FUG00250
GO TO 100	FUG00260
	FUG00270
	FUG00280
MU NE 1.0	FUG00290
	FUG00300
50 RM=1.0-1.0/MU	FUG00310
IF(TAU1.NE.0.0) GO TO 60	FUG00320
G2=EXP(TAU2/MU)*(TAU2-MU)*E1(TAU2)	FUG00330
G2=G2+MU*E1(TAU2*RM)	FUG00340
G2=G2+1.0+MU*ALOG(ABS(RM))-EXP(-RM*TAU2)	FUG00350
GO TO 100	FUG00360
60 IF(TAU2.NE.0.0) GO TO 70	FUG00370
G2=EXP(TAU1/MU)*(MU-TAU1)*E1(TAU1)	FUG00380
G2=G2-MU*E1(TAU1*RM)	FUG00390
G2=G2+EXP(-RM*TAU1)-MU*ALOG(ABS(RM))-1.0	FUG00400
GO TO 100	FUG00410
70 G2=EXP(TAU1/MU)*(MU-TAU1)*E1(TAU1)	FUG00420
G2=G2-EXP(TAU2/MU)*(MU-TAU2)*E1(TAU2)	FUG00430
G2=G2+(EXP(-TAU1*RM)-EXP(-TAU2*RM))	FUG00440
G2=G2+MU*(E1(TAU2*RM)-E1(TAU1*RM))	FUG00450
100 RETURN	FUG00460
END	FUG00470

```

FUNCTION E1(X)
COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NOIRTU,NCLIMT,KSTOR,NPLOTUF
*****
FUNCTION COMPUTES THE VALUE OF THE FIRST EXPONENTIAL INTEGRAL
-E1(-X) WHERE -174 < X < 170. FOR X OUTSIDE THESE BOUNDS, AN
OVERFLOW OR UNDERFLOW MIGHT OCCUR, SO PROGRAM EXECUTION IS
HALTED. POLYNOMIAL AND RATIONAL FUNCTION APPROXIMATIONS ARE
ADAPTED FROM THE IBM SCIENTIFIC SUBROUTINE PACKAGE, SUBROUTINE
EXPI.
*****
DATA GAMMA/.5772156649/
DATA C1,C2,C3,C4,C5,C6,C7,C8/674.567029,57.411833,6.05529232,
11699.06552,841.654932,49.3133893,8.01957683,.99979204/
DATA D1,D2,D3,D4,D5,D6,D7,D8,D9/248.6697,224.4234,32.43665,
13.061037,.05176245,180.7837,22.63818,38.93944,3.995161/
DATA F1,F2,F3,F4,F5,F6,F7,F8,F9/9.999999E-1,2.500001E-1,
15.555682E-2,1.041576E-2,1.664156E-3,2.335379E-4,2.928433E-5,
21.766345E-6,7.122452E-7/
DATA G1,G2,G3,G4,G5,G6,G7,G8/2677737343,8.6347608925,
118.059016973,8.5733287401,3.9584969228,21.0996530827,
225.6329561486,9.5733223454/
IF(X.GT.-174.0.AND.X.LE.170.0) GO TO 10
WRITE(IOOUT,1000)
STOP
10 IF(X.GT.-9.0) GO TO 20
E1=1.0-(C1+C2*X-C3*X*X-X*X*X)/(C4+C5*X+C6*X*X-C7*X*X*X-C8*X*X*X*X)
E1=E1*EXP(-X)/X
GO TO 100
20 IF(X.GT.-3.0) GO TO 30
E1=D1+D2*X+D3*X*X+D4*X*X*X+D5*X*X*X*X
E1=(1.0-E1/(D6+D7*X+D8*X*X+D9*X*X*X+X*X*X*X))*EXP(-X)/X
GO TO 100
30 IF(X.GT.1.0) GO TO 40
E1=F1-X*(F2-X*(F3-X*(F4-X*(F5-X*(F6-X*(F7-X*(F8-X*F9))))))
E1=X*E1-GAMMA-ALOG(ABS(X))
GO TO 100
40 E1=(G1+X*(G2+X*(G3+X*(G4+X))))/(G5+X*(G6+X*(G7+X*(G8+X)))
E1=E1*EXP(-X)/X
100 RETURN
1000 FORMAT(4H0X =,3X,E10.4,39H OUT-OF-RANGE FOR E1. EXECUTION HALTED.)
END

```



```

SUBROUTINE GRNADE(WAVE1,ICLMAT,TRANS,IERR)
C/*****
C/***** SUBROUTINE GRNADE
C/***** MAIN GRNADE MODULE
C/***** EOSAEL80
C/*****
DIMENSION XA(3),XTRAN(7)
COMMON /MO5/ XDTA(1000),CDTA(1000),CL(1000)
COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDECK
COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
COMMON /CLYMAT/TEMP,PRESS,RH,AH,DP,VIS,CLDMAT,CLDHYT,FOGPRB,
* WNDVEL,WNDDIR,IPASCT
COMMON /MECH0/XM,YM,ZM,XO,YO,ZO,XT,YT,ZT,ISTO,IETO,IDTO,XN,FW,
* TBURN,ITYPE,EFF,YF,RHA,UW,WD,ICAT,AIRT,TGRAD,BRATE,HEAD,RNG,
* DLEN,WPOWR,EXTC(8),XMIS(8),XNORTH
COMMON /MECH1/XI(1),YI(1),ZI(1),TTI(1),EMUN,BREXP
COMMON /MECH2/U2,WDA,THETA,UBXB,QLENTH
COMMON /MECH3/ZDIFF,YDIFF,SIGZR,XREFZ,WINDP,HK,VS,RC,HM
C/*****
C** THE FOLLOWING VARIABLES SUPPLIED BY THE USER.
C** FIELD DATA
C** XNORTH FIELD COORDINATES FROM NORTH (DEGREES)
C** HEAD GRENADE TANK HEADING CLOCKWISE FROM NORTH (DEGREES)
C** RNG DISTANCE OF GRENADES FROM TANK (METERS)
C** DLEN GRENADE SPACING (PERPENDICULAR TO HEADING)
C** XO,YO,ZO COORDINATES OF OBSERVER (METERS)
C** XM,YM,ZM COORDINATES OF TANK (METERS)
C** XT,YT,ZT TARGET COORDINATES (METERS)
C** METEOROLOGICAL DATA
C** WINDP WIND PROFILE EXPONENT
C** HM HEIGHT OF INVERSION LAYER
C** WD WIND DIRECTION FROM NORTH (DEGREES)
C** WS WIND SPEED (METERS PER SECOND)
C** RH RELATIVE HUMIDITY
C** ICAT PASQUILL CATEGORY
C** YF SMOKE YIELD FACTOR
C** MUNITION DATA
C** EFF CLOUD-MAKING EFFICIENCY OF MUNITION
C** QMUN TOTAL MASS OF SMOKE AGENT (GRAMS)
C** DETECTOR DATA
C** WAVE1 WAVELENGTH OF INTEREST (MICRONS)
C** DIFFUSION PARAMETERS
C** SIGZ REFERENCE SIGMA (METERS)
C** XREF REFERENCE DISTANCE (METERS)
C** ZDIFF VERTICAL DIFFUSION CONSTANT
C** YDIFF CROSSWIND DIFFUSION CONSTANT
C** HK TERRAIN SCAVENGING COEFFICIENT
C** VS PARTICLE SETTLING VELOCITY (CM/SEC)
C** RC TERRAIN REFLECTION COEFFICIENT
C/*****
C** DEFINITIONS OF OTHER VARIABLES**
C** IXMAX NUMBER OF POINTS ALONG LINE-OF-SIGHT FOR CL COMPUTATION
C** NBPT NUMBER OF GRENADE LINES (NBPT=1)
C** NTARG NUMBER OF TARGETS (NTARG=1)
C** TBURST MUNITION DETONATION TIME
C/*****
C** ERROR CODES AND OPTION CODES:
C** IWRIT=1 DEPRESSES RAW DATA PRINTOUT
C** IFLAG=4 INVALID DATA CARD (BUT IGNORED)
C** IFLAG=3 OVER 11 DATA CARDS ENTERED BEFORE GO
C** REMAINDER IGNORED
C** IFLAG=2 NORMAL READ TERMINATION
C** IFLAG=1 WAVELENGTH OF INTEREST NOT IN DEFINED BANDS
C** TRANS SET TO 1.0
C/*****SET DEFAULTS*****
TBURST=0.0
CALL COGET(WAVE1,KWAVE)
C/*****READ DATA AND WRITE HEADING*****
WRITE(IOOUT,8000)
8000 FORMAT(1H0,20X,40(2H**),/,21X,1H*,34X,14HPROGRAM GRNADE,30X,1H*,/,

```

21X,1H,37X,8HE0SAEL80,33X,1H*,/,21X,40(2H**))	GRN00710
IWRIT=1	GRN00720
IFLAG=0	GRN00730
6 CALL DATRD(IWRIT,IFLAG)	GRN00740
IERR=IFLAG	GRN00750
IF(IFLAG.EQ.4)GO TO 9999	GRN00760
C*****CALCULATE INTEGRATION INCREMENT	GRN00770
XX=(XT-X0)**2+(YT-Y0)**2+(ZT-Z0)**2	GRN00780
DLOS=SQRT(XX)	GRN00790
IXMAX=IFIX(DLOS)	GRN00800
IF(IXMAX.GT.1000)IXMAX=1000	GRN00810
C*****DEFAULT TO CLIMATE DATA OPTION	GRN00820
IF(ICLMAT.NE.1)GO TO 12	GRN00830
RHA=RH	GRN00840
UW=WINDVEL	GRN00850
WD=WINDDIR	GRN00860
ICAT=IFASCT	GRN00870
12 CONTINUE	GRN00880
IF(DLEN.EQ.0.0)DLEN=10.0	GRN00890
IF(BRATE.EQ.0.0)BRATE=(1.0/14.3)	GRN00900
QLENT=XN*DLEN	GRN00910
BREXP=BRATE	GRN00920
QMUN=XN*FW	GRN00930
EMUN=QMUN*YF*EFF/100.0	GRN00940
WS=UW	GRN00950
IF(XMIS(1).LE.0.0)GO TO 4	GRN00960
XREFZ=100.0	GRN00970
SIGZR=XMIS(1)	GRN00980
ZDIFF=XMIS(2)	GRN00990
YDIFF=XMIS(3)	GRN01000
HM=XMIS(4)	GRN01010
HK=XMIS(5)	GRN01020
RC=XMIS(6)	GRN01030
VS=XMIS(7)	GRN01040
GO TO 5	GRN01050
4 CALL PARMS(ICAT)	GRN01060
5 CONTINUE	GRN01070
C*****REDEFINE WIND PROFILE EXPONENT IF READ IN POSITIVE	GRN01080
IF(WPOWR.GE.0.0)WINDP=WPOWR	GRN01090
IF(WS.LE.0.0) WS=0.1	GRN01100
U2=WS	GRN01110
C-----	GRN01120
C LOCATE GENERATING LINE	GRN01130
C-----	GRN01140
THETA=(HEAD-WD)*PIRAD	GRN01150
C*** LOCATE CENTER OF GENERATING LINE.	GRN01160
YHEAD=(XNORTH+90.0-HEAD)*PIRAD	GRN01170
XBURST=RNC*SIN(YHEAD)+XM	GRN01180
YBURST=RNC*COS(YHEAD)+YM	GRN01190
ZBURST=ZM	GRN01200
C-----	GRN01210
C TRANSFORM TO OBSERVER COORDINATES	GRN01220
C-----	GRN01230
C*** TRANSFORM TO ORIGIN UNDER OBSERVER AND X-AXIS UNDER TARGET.	GRN01240
ANGLR=ATAN2(YT-Y0,XT-X0)	GRN01250
XI(1)=(XBURST-X0)*COS(ANGLR)+(YBURST-Y0)*SIN(ANGLR)	GRN01260
YI(1)=(YBURST-Y0)*COS(ANGLR)-(XBURST-X0)*SIN(ANGLR)	GRN01270
ZI(1) = ZBURST	GRN01280
TI(1) = TBURST	GRN01290
C*** GET THE WIND DIRECTION ANGLE WITH THE NEW X-AXIS.	GRN01300
ANGLD = ANGLR*180./PI	GRN01310
WX=-(WD+XNORTH+ANGLD)	GRN01320
WX=AMOD(WX,360.0)	GRN01330
WDA = WX*PI/180.	GRN01340
C-----	GRN01350
C WRITE INPUT DATA AND HEADINGS	GRN01360
C-----	GRN01370
WRITE(IOOUT,8001)	GRN01380
WRITE(IOOUT,1000)	GRN01390
WRITE(IOOUT,1001)WS,WD,ICAT,RHA,XNORTH	GRN01400

```

WRITE(1000,998)
WRITE(1000,995)
WRITE(1000,997)X0,EXTC(1),Y0,EXTC(2),Z0,EXTC(3),
*XT,EXTC(4),YT,EXTC(5),ZT,EXTC(6),EXTC(7)
WRITE(1000,1002)
WRITE(1000,1003)XM,SIGZR,YM,XREFZ,ZM,HM
WRITE(1000,1004)HEAD,HK,RNG,RC,XN,VS
WRITE(1000,1005)QMUH,WINDP,QLENTH,ZDIFF,BREXP,YDIFF
WRITE(1000,1006)EFF,YF
8001 FORMAT(//,21X,36H*****INPUT*****ALL LENGTHS IN METERS,/,
*21X,38(2H--))
1000 FORMAT(21X,15HMETEOROLOGICAL: )
1001 FORMAT(24X,10HWIND SPEED,10X,F6.1,1X,3HM/S,/,
* 24X,14HWIND DIRECTION,6X,F6.1,1X,3HDEG,/,
* 24X,17HPASQUILL CATEGORY,3X,I3,/,
* 24X,17HRELATIVE HUMIDITY,3X,F6.1,1X,1H%,/,
* 24X,21HNOTE: X AXIS HEADING,1X,F6.1,1X,3HDEG,1X,
* 28HCLOCKWISE FROM NORTH (DCWFN))
1002 FORMAT(20X,19HTANK/MUNITION DATA,16X,21HDIFFUSION PARAMETERS: )
1003 FORMAT(24X,7HX(TANK),12X,F6.1,10X,10HSIGZ(XREF),21X,F6.1,/,
* 24X,7HY(TANK),12X,F6.1,10X,4HXREF,27X,F6.1,/,
* 24X,7HZ(TANK),12X,F6.1,10X,17HMXING HEIGHT(HM),14X,F6.1)
1004 FORMAT(24X,14HHEADING(DCWFN),5X,F6.1,10X,20HSCAVENGING COEFF(HK),
*11X,F6.3,/,24X,5HRANGE,14X,F6.1,10X,20HREFLECTION COEFF(RC),11X,
*F6.3,/,24X,10HNO GRNADES,9X,F6.1,10X,21HSETTLING VELOCITY(VS),
*5X,F6.3,1X,4HCM/S)
1005 FORMAT(24X,14HSMOKE MASS(GM),5X,F6.1,10X,29HVERTICAL WIND EXPONENT
*(WPOWR),2X,F6.3,/,24X,11HLINE LENGTH,8X,F6.1,10X,
*29HVERTICAL DIFF CONSTANT(ZDIFF),2X,F6.3,/,
*24X,13HBURN CONSTANT,2X,F6.3,1X,3H1/S,10X,
*30HCROSSWIND DIFF CONSTANT(YDIFF),1X,F6.3)
1006 FORMAT(24X,10HEFFICIENCY,9X,F6.1,10X,12HYIELD FACTOR,17X,F6.1)
998 FORMAT(21X,16HOBSERVER/TARGET,8X,24HEXTINCTION COEFFICIENTS: )
995 FORMAT(49X,7HMICRONS,3X,7HM**2/GM)
997 FORMAT(24X,6HX(OBS),3X,F6.1,10X,7H0.4-0.7,3X,F6.3,/,
* 24X,6HY(OBS),3X,F6.1,10X,7H0.7-1.2,3X,F6.3,/,
* 24X,6HZ(OBS),3X,F6.1,10X,7H1.06,3X,F6.3,/,
* 24X,6HX(TAR),3X,F6.1,10X,7H3.0-5.0,3X,F6.3,/,
* 24X,6HY(TAR),3X,F6.1,10X,7H8.0-12,3X,F6.3,/,
* 24X,6HZ(TAR),3X,F6.1,10X,7H10.6,3X,F6.3,/,
* 49X,4H94.0,1X,3HGHZ,2X,F6.3)
WRITE(1000,996)
996 FORMAT(1H1,21X,16H*****OUTPUT*****/,21X,38(2H--))
WRITE(1000,3000)
3000 FORMAT(24X,4HTIME,6X,2HCL,23X,12HTRANSMISSION,/,24X,5H(SEC),2X,
*9H(GM/M**2),2X,7H0.4-0.7,1X,7H0.7-1.2,3X,4H1.06,2X,7H3.0-5.0,1X,
*7H8.0-12,2X,4H10.6,4X,5H94GHZ)
C-----
C BEGIN CL CALCULATIONS
C-----
DO 400 IT=ISTO,IETO,IDTO
ITT=IT
C * SET UP LOOP ON SPACIAL DISTRIBUTION
XC = 0.0
YC = 0.0
ZC = Z0
DELX=SQRT((XT-X0)**2+(YT-Y0)**2)/IXMAX
DELZ = (ZT-Z0)/IXMAX
C** FOR EACH TIME GET THE CONCENTRATION AT IXMAX POINTS ALONG LINE-OF-SIGHT
DO 300 IX=1,IXMAX
XC = XC + DELX
ZC = ZC + DELZ
XA(1) = XC
XA(2) = YC
XA(3) = ZC
T = FLOAT(IT)
UBXB=U2
CALL CONCN(XA,T,C)
XDTA(IX)=XC
CDTA(IX)=C

```

GRN01410
 GRN01420
 GRN01430
 GRN01440
 GRN01450
 GRN01460
 GRN01470
 GRN01480
 GRN01490
 GRN01500
 GRN01510
 GRN01520
 GRN01530
 GRN01540
 GRN01550
 GRN01560
 GRN01570
 GRN01580
 GRN01590
 GRN01600
 GRN01610
 GRN01620
 GRN01630
 GRN01640
 GRN01650
 GRN01660
 GRN01670
 GRN01680
 GRN01690
 GRN01700
 GRN01710
 GRN01720
 GRN01730
 GRN01740
 GRN01750
 GRN01760
 GRN01770
 GRN01780
 GRN01790
 GRN01800
 GRN01810
 GRN01820
 GRN01830
 GRN01840
 GRN01850
 GRN01860
 GRN01870
 GRN01880
 GRN01890
 GRN01900
 GRN01910
 GRN01920
 GRN01930
 GRN01940
 GRN01950
 GRN01960
 GRN01970
 GRN01980
 GRN01990
 GRN02000
 GRN02010
 GRN02020
 GRN02030
 GRN02040
 GRN02050
 GRN02060
 GRN02070
 GRN02080
 GRN02090
 GRN02100

300 CONTINUE	GRN02110
C*** INTEGRATE ALONG LINE-OF-SIGHT TO GET CL.	GRN02120
CALL SUMA (XDTA,CDTA, CL ,IXMAX)	GRN02130
C-----	GRN02140
C CALCULATE TRANSMITTANCE	GRN02150
C-----	GRN02160
DO 500 J=1,7	GRN02170
XTRAN(J)=EXP(-1.0*EXTC(J)*CL(IXMAX))	GRN02180
500 CONTINUE	GRN02190
WRITE(100OUT,2000)T,CL(IXMAX),(XTRAN(J),J=1,7)	GRN02200
2000 FORMAT(23X,F5.1,2X,F8.3,2X,7(1X,F7.3))	GRN02210
400 CONTINUE	GRN02220
C*****	GRN02230
C*****SET TRANSMISSION FOR RETURN TO EOSAEL	GRN02240
C*****	GRN02250
IF(KWAVE.GT.0)GO TO 3	GRN02260
TRANS=1.0	GRN02270
IFLAG=1	GRN02280
3 TRANS=XTRAN(KWAVE)	GRN02290
GO TO 6	GRN02300
9999 WRITE(100OUT,9000)	GRN02310
9000 FORMAT(21X,28H*****PROGRAM GRNADE END*****)	GRN02320
RETURN	GRN02330
END	GRN02340

```

SUBROUTINE CONCNC(XA,T,C2)
C/*****
C/ SUBROUTINE CONCNC
C/ GRNAD MODULE
C/ EOSAEL80
C/*****
C/ PURPOSE:
C/ CALCULATES CONCENTRATION AT A SPECIFIED POSITON AND TIME.
C/ USAGE:
C/ THE CONCENTRATION IS USED WITH THE EXTINCTION COEFFICIENT
C/ TO COMPUTE TRANSMITTANCE.
C/ DESCRIPTION OF PARAMETERS:
C/ XA - POSITION IN METERS, INPUT.
C/ T - TIME IN SECONDS, INPUT.
C/ C2 - CONCENTRATION, OUTPUT.
C/ SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED:
C/ LOCAT UMEAN
C/ COMMON BLOCK STATEMENTS REQUIRED:
C/ MECH1 MECH2 MECH3
C/ REMARKS:
C/ CONCNC COMPUTES FIVE TERMS AND MULTIPLIES THEM TO
C/ GET CONCENTRATION.
C/*****
C/ DIMENSION XA(3)
C/ COMMON/CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDECK
C/ COMMON/MECH1/XI(1),YI(1),ZI(1),TTI(1),EMUN,BREXP
C/ COMMON/MECH2/U2,WDA,THETA,UBXB,QLENTH
C/ COMMON/MECH3/ZDIFF,YDIFF,SIGZR,XREFZ,WINDP,HK,VS,RC,HM
C-----
C/ CONCENTRATION=TERM1*TERM2*TERM3*TERM4*TERMS
C-----
C/ C2 = 0.0
C/ X = XA(1)
C/ Y = XA(2)
C/ Z = XA(3)
C*** GET CROSSWIND AND DOWNWIND COMPONENTS OF GENERATING LINE.
C/ QC=ABS(QLENTH*COS(THETA))
C/ QD=ABS(QLENTH*SIN(THETA))
C/ NBPT=1
C/ DO 390 J = 1,NBPT
C/ TJ = T- TTI(J)
C-----
C/ TERM1 IS CLOUD MASS AS A FUNCTION OF TIME.
C-----
C/ IF(BREXP*TJ.GE.200.0)STOP01
C/ TRM1=EMUN*(1.0-EXP(-BREXP*TJ))
C-----
C/ TERM2 IS TERRAIN SCAVENGING TERM.
C-----
C** CHANGE TO SMOKE COORDINATES.
C/ CALL LOCAT(J,X,Y,Z,XB,YB,ZB)
C*** UPWIND END OF GENERATING LINE WILL BE ORIGIN OF SMOKE SYSTEM.
C/ XB=XB+0.5*QD
C/ CALL UMEAN(J,TJ)
C/ IF(HK*XB/UBXB.GE.200.0)STOP01
C/ TRM2 = EXP(-HK*XB/UBXB)
C-----
C/ TERM3 IS DOWNWIND PROBABILITY DENSITY.
C-----
C/ UT=UBXB*TJ
C/ IF(XB.LE.0.) GO TO 999
C/ IF(XB.GT.0. AND.XB.LE.UT) TERM3=1./(UT+0.5*QD)
C/ IF(XB.GT.UT AND.XB.LE.UT+QD) TERM3=(UT+QD-XB)/(UT*QD+0.5*QD*QD)
C/ IF(XB.GT.UT+QD) TERM3=0.
C/ TRM3=TERM3
C-----
C/ TERM4 IS CROSSWIND PROBABILITY DENSITY.
C-----
C/ YWIDTH=YDIFF*XB+QC
C/ TRM4= 1./YWIDTH

```

```

COC00010
COC00020
COC00030
COC00040
COC00050
COC00060
COC00070
COC00080
COC00090
COC00100
COC00110
COC00120
COC00130
COC00140
COC00150
COC00160
COC00170
COC00180
COC00190
COC00200
COC00210
COC00220
COC00230
COC00240
COC00250
COC00260
COC00270
COC00280
COC00290
COC00300
COC00310
COC00320
COC00330
COC00340
COC00350
COC00360
COC00370
COC00380
COC00390
COC00400
COC00410
COC00420
COC00430
COC00440
COC00450
COC00460
COC00470
COC00480
COC00490
COC00500
COC00510
COC00520
COC00530
COC00540
COC00550
COC00560
COC00570
COC00580
COC00590
COC00600
COC00610
COC00620
COC00630
COC00640
COC00650
COC00660
COC00670
COC00680
COC00690
COC00700

```

	IF(YB.GT. 0.5*YWIDTH) TRM4=0.	COC00710
	IF(YB.LT. -0.5*YWIDTH) TRM4=0.	COC00720
C	-----	COC00730
C	TERMS IS VERTICAL PROBABILITY DENSITY.	COC00740
	-----	COC00750
	SIGZT=SIGZR*(XB/XREFZ)**ZDIFF	COC00760
	SZT2=SIGZT*SIGZT	COC00770
	HMVT = ZI(J) - (VS/100.0)*(XB/UBXB)	COC00780
	TV1 = EXP(-(HMVT-ZB)**2/(2.0*SZT2))	COC00790
	TV2 = RC*EXP(-(HMVT+ZB)**2/(2.0*SZT2))	COC00800
	TERR = 1.0E-06	COC00810
	JC = 0	COC00820
	T1 = 0.0	COC00830
	T2 = 0.0	COC00840
	T3 = 0.0	COC00850
	T4 = 0.0	COC00860
	TS = 0.0	COC00870
370	JC = JC+1	COC00880
	R1 = RC**JC	COC00890
	R2 = RC**JC	COC00900
	R3 = RC**JC	COC00910
	IF((2.*JC*HM+HMVT+ZB)**2.GT. 600.*2.*SZT2) GO TO 371	COC00920
	T1 = R1*EXP(-(2.0*JC*HM-HMVT-ZB)**2/(2.0*SZT2))	COC00930
	T2 = R2*EXP(-(2.0*JC*HM-HMVT+ZB)**2/(2.0*SZT2))	COC00940
	T3 = R3*EXP(-(2.0*JC*HM+HMVT-ZB)**2/(2.0*SZT2))	COC00950
	T4 = R3*EXP(-(2.0*JC*HM+HMVT+ZB)**2/(2.0*SZT2))	COC00960
	T1234 = T1 + T2 + T3 + T4	COC00970
	TS = TS + T1234	COC00980
	IF(T1234 - TERR) 371,371,370	COC00990
371	CONTINUE	COC01000
	TV3 = TS	COC01010
	TRM5 = TV1+TV2+TV3	COC01020
	TRM5=TRM5/(SIGZT*SQRT(2.*PI))	COC01030
	C1 = TRM1*TRM2*TRM3*TRM4*TRM5	COC01040
	C2 = C2 + C1	COC01050
380	CONTINUE	COC01060
999	RETURN	COC01070
	END	COC01080

```

      SUBROUTINE GOGET(WAVE1,KWAVE)
C/*****
C/*          SUBROUTINE GOGET
C/*          GRNAD MODULE
C/*          EOSAEL80
C/*****
C*****SUBROUTINE FINDS SPECTRAL BAND FOR GIVEN SINGLE WAVELENGTH
      KWAVE=0
      IF(WAVE1.GE.0.40.AND.WAVE1.LT.0.70)KWAVE=1
      IF(WAVE1.GE.0.70.AND.WAVE1.LT.1.20)KWAVE=2
      IF(WAVE1.GE.1.20.AND.WAVE1.LT.3.00)KWAVE=3
      IF(WAVE1.GE.3.00.AND.WAVE1.LT.5.00)KWAVE=4
      IF(WAVE1.GE.5.00.AND.WAVE1.LT.12.0)KWAVE=5
      IF(WAVE1.EQ.1.06)KWAVE=3
      IF(WAVE1.EQ.10.6)KWAVE=6
      IF(WAVE1.EQ.94.0)KWAVE=7
      IF(WAVE1.GT.3188.0.AND.WAVE1.LT.3195.0)KWAVE=7
      RETURN
      END

```

```

GOG00010
*/GOG00020
*/GOG00030
*/GOG00040
*/GOG00050
*/GOG00060
GOG00070
GOG00080
GOG00090
GOG00100
GOG00110
GOG00120
GOG00130
GOG00140
GOG00150
GOG00160
GOG00170
GOG00180

```

```

SUBROUTINE PARM5(ICAT)
C/*****
C/* SUBROUTINE PARM5
C/* GRNAD MODULE
C/* EOSAEL80
C/*****
C*****SETS DIFFUSION PARAMETER DEFAULTS AS FUNCTION OF PASQUILL CATEGORY
COMMON/MECH3/ZDIFF,YDIFF,SIGZR,XREFZ,WINDP,HK,VS,RC,HM
YDIFF=0.355
HK=0.002
VS=0.021
RC=0.70
XREFZ=100.0
IF(ICAT.GT.3)GO TO 1
WINDP=0.10
ZDIFF=2.08
HM=1000.0
SIGZR=14.0
GO TO 3
1 IF(ICAT.GT.4)GO TO 2
WINDP=0.20
ZDIFF=1.40
HM=300.0
SIGZR=7.2
GO TO 3
2 WINDP=0.40
ZDIFF=1.04
HM=50.0
SIGZR=5.0
3 RETURN
END
PAR00010
PAR00020
PAR00030
PAR00040
PAR00050
PAR00060
PAR00070
PAR00080
PAR00090
PAR00100
PAR00110
PAR00120
PAR00130
PAR00140
PAR00150
PAR00160
PAR00170
PAR00180
PAR00190
PAR00200
PAR00210
PAR00220
PAR00230
PAR00240
PAR00250
PAR00260
PAR00270
PAR00280
PAR00290
PAR00300
PAR00310

```


SUBROUTINE EXTIN(EX)		EXTN0010
C/*****		EXTN0020
C/*	SUBROUTINE EXTIN	*/EXTN0030
C/*	GRNAD MODULE	*/EXTN0040
C/*	EOSAEL80	*/EXTN0050
C/*****		EXTN0060
C*****PROGRAM TO SET EXTINCTION COEFFICIENTS FOR WP/RP SMOKE****		EXTN0070
	DIMENSION EX(7),CX(7)	EXTN0080
	DATA CX/4.304,2.166,1.541,0.350,0.338,0.364,0.001/	EXTN0090
	DO 1 I=1,7	EXTN0100
	EX(I)=CX(I)	EXTN0110
1	CONTINUE	EXTN0120
	RETURN	EXTN0130
	END	EXTN0140

```

SUBROUTINE UMEAN(J,TJ)
C/*****
C/* SUBROUTINE UMEAN
C/* GRNAD MODULE
C/* EOSAEL80
C/*****
C*****CALCULATES MEAN WIND SPEED OVER EXTENT OF CLOUD
COMMON/MECH1/XI(1),YI(1),ZI(1),TTI(1),EMUN,BREXP
COMMON/MECH2/U2,WDA,THETA,UBXB,QLENTH
COMMON/MECH3/ZDIFF,YDIFF,SIGZR,XREFZ,WINDP,HK,VS,RC,HM
IC = 0
QD = ABS(QLENTH*SIN(THETA))
P = WINDP + 1.0
C1=P*2.0**WINDP
Z2=ZI(J)
IF(Z2-2.0)10,10,20
10 CONTINUE
Z2=2.0
Z1=0.01
GO TO 200
20 Z1=ZI(J)-1.5*SIGZR
IF(Z1)30,30,40
30 Z1=0.01
40 CONTINUE
UH = (U2/((Z2-Z1)*C1))*(Z2**P-Z1**P)
US=UH
50 CONTINUE
XC = UH*TJ
Z2 = ZI(J) - (VS/100.0)*(XC/UH)
IF(Z2-2.0)151,151,155
151 CONTINUE
Z2=2.0
Z1=0.01
GO TO 160
155 CONTINUE
SIGZT = SIGZR*((XC+QD)/XREFZ)**ZDIFF
Z1 = ZI(J) - 1.5*SIGZT
IF(Z1)159,159,160
159 CONTINUE
Z1=0.01
160 UBXJ = (U2/((Z2-Z1)*C1))*(Z2**P-Z1**P)
GO TO 1000
200 CONTINUE
US=U2
UH=US
UBXJ=US
1000 CONTINUE
UBXJ = SQRT((UBXJ**2 + UH**2)/2.0)
IC = IC + 1
IF(IC,EQ,1) UH = UBXJ
IF(IC,EQ,1) GO TO 50
UBXB=UBXJ
RETURN
END

```

```

/UME00010
/UME00020
*/UME00030
*/UME00040
*/UME00050
/UME00060
/UME00070
/UME00080
/UME00090
/UME00100
/UME00110
/UME00120
/UME00130
/UME00140
/UME00150
/UME00160
/UME00170
/UME00180
/UME00190
/UME00200
/UME00210
/UME00220
/UME00230
/UME00240
/UME00250
/UME00260
/UME00270
/UME00280
/UME00290
/UME00300
/UME00310
/UME00320
/UME00330
/UME00340
/UME00350
/UME00360
/UME00370
/UME00380
/UME00390
/UME00400
/UME00410
/UME00420
/UME00430
/UME00440
/UME00450
/UME00460
/UME00470
/UME00480
/UME00490
/UME00500
/UME00510
/UME00520
/UME00530
/UME00540

```

```

SUBROUTINE SUMA(X,Y,Z,NDIM)
C/*****
C/ SUBROUTINE SUMA
C/ GRNAD MODULE
C/ EOSAEL80
C/*****
C/ PURPOSE:
C/ GENERAL PURPOSE INTEGRATION SUBROUTINE
C/ USAGE:
C/ CALLED FROM GRNAD TO GET INTEGRAL OF CONCENTRATION ALONG
C/ LINE-OF-SIGHT.
C/ DESCRIPTION OF PARAMETERS:
C/ X INDEPENDENT VARIABLE
C/ Y DEPENDENT VARIABLE
C/ Z INTEGRAL OF Y OVER X
C/ INDIM NUMBER OF POINTS
C/ SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED:
C/ NONE
C/ COMMON BLOCK STATEMENTS REQUIRED:
C/ NONE
C/ REMARKS:
C/ AS USED BY GRENADE, X IS POSITION ONLINE-OF-SIGHT, Y IS
C/ LOCAL CONCENTRATION, Z IS TOTAL CONCENTRATION.
C/ METHOD:
C/ SUMA INCREMENTS INTEGRAL BY AVERAGE OF LAST TWO Y VALUES
C/ TIMES DELTA X.
C/*****
      DIMENSION X(1000),Y(1000),Z(1000)
      SUM2 = 0.0
      IF (NDIM - 1) 4,3,1
      *** INTEGRATION LOOP ***
1      DO 2 I = 2,NDIM
          SUM1 = SUM2
          SUM2 = SUM2 + 0.500*(X(I)-X(I-1))*(Y(I)+Y(I-1))
2      Z(I-1) = SUM1
3      Z(NDIM) = SUM2
4      RETURN
      END

```

```

SUMA0010
SUMA0020
SUMA0030
SUMA0040
SUMA0050
SUMA0060
SUMA0070
SUMA0080
SUMA0090
SUMA0100
SUMA0110
SUMA0120
SUMA0130
SUMA0140
SUMA0150
SUMA0160
SUMA0170
SUMA0180
SUMA0190
SUMA0200
SUMA0210
SUMA0220
SUMA0230
SUMA0240
SUMA0250
SUMA0260
SUMA0270
SUMA0280
SUMA0290
SUMA0300
SUMA0310
SUMA0320
SUMA0330
SUMA0340
SUMA0350
SUMA0360
SUMA0370
SUMA0380
SUMA0390

```

```

SUBROUTINE LOCAT(I,X,Y,Z,XB,YB,ZB)
C/*****
C/* SUBROUTINE LOCAT
C/* GRNAD MODULE
C/* EOSAEL80
C/*****
C/* PURPOSE:
C/* LOCAT TRANSLATES POSITION INTO BURST AND WIND COORDINATES.
C/* USAGE:
C/* CALLED BY CONCH
C/* DESCRIPTION OF PARAMETERS:
C/* T TIME (SECONDS), INPUT
C/* X,Y,Z ORIGINAL POSITION (METERS), INPUT
C/* XI,YI,ZI ORIGINAL BURST POSITION (METERS), INPUT
C/* WDA ANGLE BETWEEN WIND VECTOR AND LINE OF SIGHT
C/* (RADIAN), INPUT
C/* XB,YB,ZB TRANSLATED POSITION (METERS), OUTPUT
C/* SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED:
C/* NONE
C/* COMMON BLOCK STATEMENTS REQUIRED:
C/* MECH1 MECH2
C/* REMARKS:
C/* ORIGINAL COORDINATES HAVE ORIGIN AT OBSERVER AND X-AXIS
C/* THROUGH TARGET. NEW COORDINATES HAVE ORIGIN AT BURST
C/* AND X-AXIS IN DIRECTION OF WIND VECTOR.
C/* METHOD:
C/* STANDARD ROTATION AND TRANSLATION OF AXES
C/*****
COMMON/MECH1/XI(1),YI(1),ZI(1),TTI(1),EMUN,BREXP
COMMON/MECH2/U2,WDA,THETA,UBXB,QLENTH
XB = (X - XI(1))*COS(WDA) + (Y - YI(1))*SIN(WDA)
YB = -(X - XI(1))*SIN(WDA) + (Y - YI(1))*COS(WDA)
ZB = Z
IF(ZB.LT.0.0.AND.ABS(ZB).GT.ZI(1)) ZB=0.0
RETURN
END

```

```

LOC00010
LOC00020
LOC00030
LOC00040
LOC00050
LOC00060
LOC00070
LOC00080
LOC00090
LOC00100
LOC00110
LOC00120
LOC00130
LOC00140
LOC00150
LOC00160
LOC00170
LOC00180
LOC00190
LOC00200
LOC00210
LOC00220
LOC00230
LOC00240
LOC00250
LOC00260
LOC00270
LOC00280
LOC00290
LOC00300
LOC00310
LOC00320
LOC00330
LOC00340
LOC00350
LOC00360

```

```

SUBROUTINE DATRD(IWRIT,IFLAG)
C/*****
C/***** SUBROUTINE DATRD
C/***** GRNAD MODULE
C/***** EOSAEL80
C/***** THIS SUBROUTINE READS INPUT DATA IN EXACTLY THE SAME FORMAT AS
C/***** THE SMOKE(EOSAEL) PROGRAM
C/*****
C/***** EACH CARD BEGINS WITH A 4 LETTER IDENTIFIER IN COL 1-4,
C/***** FOLLOWED BY AS MANY (REAL) FIELDS AS NEEDED, 10 COL.
C/***** PER FIELD BEGINNING IN COL 11. THE CARDS ARE NOT ORDER
C/***** DEPENDENT.
C/*****
C/***** NAME IGNORED
C/***** BURN IGNORED
C/***** MUNC XM,YM,ZM COORDINATES OF GRENADE FIRING TANK
C/***** HEAD HEADING OF GRENADE FIRING TANK CLOCKWISE
C/***** FROM NORTH
C/***** RNG GRENADE FIRING RANGE OF TANK
C/***** DLEN SPACING OF GRENADES ALONG LINE PERPENDICULAR
C/***** TO HEADING
C/***** OBSC XO,YO,ZO COORDINATES OF THE OBSERVER (M,M,M)
C/***** TARC XT,YT,ZT COORDINATES OF THE TARGET (M,M,M)
C/***** BART
C/***** STO STARTING TIME (ELAPSED TIME SINCE BLAST)
C/***** ETO ENDING TIME FOR CALCULATION
C/***** DTD TIME INCREMENT FOR CALCULATION
C/***** XNORTH X AXIS HEADING CLOCKWISE FROM NORTH
C/*****
C/***** MUNT
C/***** XN NUMBER OF MUNITIONS FIRED AT THE SAME
C/***** LOCATION AND AT THE SAME TIME
C/***** FW FILL WEIGHT (LBS)
C/***** TBURN BURN TIME OF SMOKE TYPE (IGNORED)
C/***** ITYPE TYPE OF SMOKE (DEFAULTS TO 1)
C/***** 1.=WP, 2.=PWP, 3.=HC, 4.=FOG OIL
C/***** EFF EFFICIENCY OF BURN (PERCENT). IF 0.0,
C/***** DEFAULTS TO 62.0%.
C/***** YF YIELD FACTOR IF 0.0,DEFAULTS TO ANALYTICAL
C/***** MODEL
C/***** BRATE EXPONENTIAL BURN RATE PARAMETER
C/*****
C/***** METR
C/***** RHA RELATIVE HUMIDITY (PERCENT)
C/***** UW WIND VELOCITY (M/S)
C/***** WD WIND DIRECTION (DEGREES)
C/***** ICAT PASQUILL CATEGORY
C/***** 1-A, 2-B, 3-C, 4-D, 5-E, 6-F
C/***** AIRT SURFACE AIR TEMPERATURE (IGNORED)
C/***** TGRAD VERT TEMP GRADIENT (IGNORED)
C/***** WPOWR WIND PROFILE EXPONENT (DIMENSIONLESS)
C/*****
C/***** EXTC
C/***** DESIRED CHANGES IN EXTINCTION COEFF.
C/***** (OPTIONAL). IF NOT USED OR READ AS 0,
C/***** DEFAULTS TO ALPHA ARRAY VALUE IN STRNS.
C/***** BANDS ARE:
C/***** 0.4-0.7 MICRONS
C/***** 0.7-1.2 MICRONS
C/***** 1.06 MICRONS
C/***** 3.0-5.0 MICRONS
C/***** 8.0-12. MICRONS
C/***** 10.6 MICRONS
C/***** 94.0 CMZ
C/*****
C/***** MISC
C/***** SIGZR DIFFUSION PARAMETER OPTION CARD FOR GRNAD
C/***** ZDIFF DOWNWIND REFERENCE AT 100 M REFERENCE DIST.
C/***** YDIFF VERTICAL DIFFUSION COEFFICIENT
C/***** HM CROSSWIND DIFFUSION COEFFICIENT
C/***** HK HEIGHT OF MIXING LAYER (METERS)
C/***** RC TERRAIN SCAVENGING COEFFICIENT
C/***** VS TERRAIN REFLECTION COEFFICIENT
C/***** SETTLE VEL (CM SEC)
C/*****
C/***** GO SIGNIFIES END OF THIS RUN, BUT NOT END OF INPUT
C/***** DONE END OF JOB.

```

```

C*****
COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUDAT00710
COMMON /GEOMET/PTS<15>,IGEOSWDAT00720
COMMON /CLYMAT/TEMP,PRESS,RH,AH,DP,VIS,CLDAMT,CLOHYT,FOGPRB,DAT00730
* WNDVEL,WNDDIR,IPASCTDAT00740
COMMON /MECH0/XM,YM,ZM,XO,YO,ZO,XT,YT,ZT,ISTO,IETO,IDTO,XN,FW,DAT00750
*TBURN,ITYPE,EFF,YF,RHA,UW,WD,ICAT,AIRT,TGRAD,BRATE,HEAD,RNG,DAT00760
*DLN,WPOWR,EXTC<8>,XMIS<8>,XNORTH,DAT00770
DIMENSION DUMY<8>,IR<26>,IR1<10>,R1<10>,EX<7>,INAME<35>DAT00780
DATA IR/2HME,2HTR,2HMU,2HNT,2HBA,2HRT,2HMU,2HNC,2HOB,2HSC,2HTA,DAT00790
*2HRC,2HEX,2HTC,2HBU,2HRN,2HMI,2HSC,2HGO,2H ,2HDO,2HNE,2HNA,2HME,DAT00800
*2HOU,2HTP/DAT00810
IF (IFLAG.GT.0) GO TO 8DAT00820
DO 2 J=1,8DAT00830
XMIS(J)=0.0DAT00840
2 EXTC(J)=0.DAT00850
C***DAT00860
C*** BEGINNING OF READ LOOPDAT00870
C***DAT00880
8 CONTINUEDAT00890
IF(IWRIT.EQ.0)GO TO 6DAT00900
WRITE(IOOUT,200)DAT00910
200 FORMAT(1H0,21X,20H*****CARD INPUT*****,/,21X,40(2H--))DAT00920
201 FORMAT(2A2,6X,35A2)DAT00930
202 FORMAT(1H0,21X,2A2,6X,35A2)DAT00940
6 DO 70 I=1,13DAT00950
IF(I.EQ.13) GO TO 310DAT00960
IF(IFLAG.GT.0) GO TO 4DAT00970
IFLAG=1DAT00980
READ(IOIN,201)IR1<1>,IR1<2>,<INAME(J),J=1,35>DAT00990
IF(IWRIT.EQ.0) GO TO 4DAT01000
WRITE(IOOUT,202)IR1<1>,IR1<2>,<INAME(J),J=1,35>DAT01010
4 READ(IOIN,20)IR1<1>,IR1<2>,<R1(J),J=2,8>DAT01020
IF(IWRIT.EQ.0) GO TO 5DAT01030
WRITE(IOOUT,30)IR1<1>,IR1<2>,<R1(J),J=2,8>DAT01040
5 IF(IR1<1>.EQ.IR<21> .AND. IR1<2>.EQ.IR<22>) GO TO 998DAT01050
20 FORMAT(2A2,6X,7F10.3)DAT01060
30 FORMAT(1H0,21X,2A2,6X,7F10.3)DAT01070
C*****DAT01080
C*** RELATING INPUT DATA TO VARIABLE NAMES.DAT01090
C*****DAT01100
IF(IR1<1>.EQ.IR<1> .AND. IR1<2>.EQ.IR<2>) GO TO 90DAT01110
IF(IR1<1>.EQ.IR<3> .AND. IR1<2>.EQ.IR<4>) GO TO 100DAT01120
IF(IR1<1>.EQ.IR<5> .AND. IR1<2>.EQ.IR<6>) GO TO 110DAT01130
IF(IR1<1>.EQ.IR<7> .AND. IR1<2>.EQ.IR<8>) GO TO 120DAT01140
IF(IR1<1>.EQ.IR<9> .AND. IR1<2>.EQ.IR<10>) GO TO 130DAT01150
IF(IR1<1>.EQ.IR<11> .AND. IR1<2>.EQ.IR<12>) GO TO 140DAT01160
IF(IR1<1>.EQ.IR<13> .AND. IR1<2>.EQ.IR<14>) GO TO 150DAT01170
IF(IR1<1>.EQ.IR<15> .AND. IR1<2>.EQ.IR<16>) GO TO 155DAT01180
IF(IR1<1>.EQ.IR<17> .AND. IR1<2>.EQ.IR<18>) GO TO 165DAT01190
IF(IR1<1>.EQ.IR<19> .AND. IR1<2>.EQ.IR<20>) GO TO 175DAT01200
IF(IR1<1>.EQ.IR<21> .AND. IR1<2>.EQ.IR<22>) GO TO 998DAT01210
IF(IR1<1>.EQ.IR<23> .AND. IR1<2>.EQ.IR<24>) GO TO 70DAT01220
IF(IR1<1>.EQ.IR<25> .AND. IR1<2>.EQ.IR<26>) GO TO 70DAT01230
C*****DAT01240
C ERROR CAUTION FOR INVALID DATA CARDDAT01250
C*****DAT01260
IFLAG=2DAT01270
WRITE(IOOUT,80)DAT01280
80 FORMAT(21X,35H*****CAUTION***** INVALID DATA CARD)DAT01290
GO TO 70DAT01300
90 RHA = R1<2>DAT01310
UW = R1<3>DAT01320
WD = R1<4>DAT01330
ICAT =IFIX(R1<5>)DAT01340
AIRT = R1<6>DAT01350
TGRAD = R1<7>DAT01360
WPOWR=R1<8>DAT01370
GO TO 70DAT01380
100 XN = R1<2>DAT01390

```

FW	=453.6*R1(3)	DAT01410
TBURN	= R1(4)	DAT01420
ITYPE	= IFIX(R1(5))	DAT01430
EFF	= R1(6)	DAT01440
YF=R1(7)		DAT01450
BRATE=R1(8)		DAT01460
GO TO 70		DAT01470
110 ISTO	= IFIX(R1(2))	DAT01480
IETO	= IFIX(R1(3))	DAT01490
IDTO	= IFIX(R1(4))	DAT01500
XNORTH= R1(5)		DAT01510
IF(ISTO.LE.0)ISTO=1		DAT01520
GO TO 70		DAT01530
120 XM	= R1(2)	DAT01540
YM	= R1(3)	DAT01550
ZM	= R1(4)	DAT01560
HEAD	= R1(5)	DAT01570
RNG	= R1(6)	DAT01580
DLEN	= R1(7)	DAT01590
GO TO 70		DAT01600
130 XO	= R1(2)	DAT01610
YO	= R1(3)	DAT01620
ZO	= R1(4)	DAT01630
GO TO 70		DAT01640
140 XT	= R1(2)	DAT01650
YT	= R1(3)	DAT01660
ZT	= R1(4)	DAT01670
C*****		DAT01680
C*****BURN CARD DATA DUMMYED BY PROGRAM GRNAD*****		DAT01690
C*****		DAT01700
155 DO 156 J=1,7		DAT01710
156 DUMY(J)=R1(J+1)		DAT01720
GO TO 70		DAT01730
150 DO 152 J=1,7		DAT01740
152 EXTC(J)=R1(J+1)		DAT01750
GO TO 70		DAT01760
165 DO 166 J=1,7		DAT01770
166 XMIS(J)=R1(J+1)		DAT01780
70 CONTINUE		DAT01790
175 GO TO 311		DAT01800
C*****		DAT01810
C*****CAUTION FOR TOO MANY CARDS		DAT01820
C*****		DAT01830
310 WRITE(IOOUT,320)		DAT01840
IFLAG=3		DAT01850
C*****		DAT01860
C*****DEFAULT NON USER DEFINED INPUT*****		DAT01870
C*****		DAT01880
311 IF(ITYPE.EQ.1)GO TO 3		DAT01890
ITYPE=1		DAT01900
WRITE(IOOUT,171)		DAT01910
171 FORMAT(1H,21X,17H*****CAUTION*****/,1H,21X,54HWRONG SMOKE TYPE		DAT01920
*FOR PROGRAM GRNAD--DEFAULTED TO WP/RP)		DAT01930
3 IF(EFF.EQ.0.0)EFF=62.0		DAT01940
IF(YF.EQ.0.0)YF=3.14+0.032*RHA		DAT01950
IF(EXTC(1).GT.0.0)GO TO 1		DAT01960
CALL EXTN(EX)		DAT01970
DO 7 I=1,7		DAT01980
7 EXTC(I)=EX(I)		DAT01990
1 CONTINUE		DAT02000
320 FORMAT(21X,17H*****CAUTION*****/,		DAT02010
*21X,56HMORE THAN 10 DATA CARDS ENTERED--REMAINING CARDS IGNORED)		DAT02020
GO TO 9999		DAT02030
998 IFLAG=4		DAT02040
9999 IF(IGEOSW.NE.1) GO TO 555		DAT02050
DISKTM=1000.		
C*** CONVERT KM TO M.		DAT02060
XT=PTS(1)*DISKTM		DAT02070
YT=PTS(2)*DISKTM		DAT02080
ZT=PTS(3)*DISKTM		

X0=PTS(4)*DISKTM
Y0=PTS(5)*DISKTM
Z0=PTS(6)*DISKTM
555 RETURN
END

DAT02090
DAT02100
DAT02110
DAT02120
DAT02130


```

SUBROUTINE LT4MCH1,H2,ANGLE,ITYPE,IXY,TRAN,RADA,RADG,IEMISS,LEN, L4M00010
+MODEL,VIS,V11,V22,T1,ICLMAT,IERR,NR,HAZE,MULDV) L4M00020
LOGICAL ISPOT,LOREAD,N16 L4M00030
COMMON /CONST/PI,PI2,CA,TWOPI,TORRMB,CDEGK L4M00040
COMMON /MO1/EH(16,34),P(34),T(34),WH(34),Z(34),WA(34),RE,M,NL L4M00050
COMMON /MO2/WO(34),RO,TBOUND,JP,IM,ML,IP,JSTOR L4M00060
COMMON /MO5/C1(501),C2(258),C3(86),C4(33),C5(6),C5DUM(9),C8(82), L4M00070
1 C11(4),C12(15),C14(21),C15(6) L4M00080
COMMON /MO7/TR(67),FW(67),FO(67) L4M00090
COMMON /MO8/SUM4,SUM5,SUM8,SUM11,SUM6 L4M00100
COMMON /MO9/RADMAX,RADMIN,VRMAX,VRMIN L4M00110
COMMON /MO3/FS(9),S1(9),S2(9),FNH3(9),FH1(9),FH2(9),FN02(9), L4M00120
1 O1(9),O2(9),PPMS02,PPMNH3,PPMNO2 L4M00130
COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUL L4M00140
COMMON /LOWEX/WPATH(68,16),WLAY(34,16),TBBY(68),TX(16),BETAEX, L4M00150
1 CLDHGT,NCLD L4M00160
COMMON /EM1/HMIN,KMAX,IJ,J1,J2,JMIN,JEXTRA,NP1 L4M00170
COMMON /EM2/W(16),E(16),IL,IKMAX,LENTOR,NLL L4M00180
COMMON /BASPT/ANG(65),SUM(65),WVL(16),NWVL,ALBB(16),BS(16), L4M00190
1 BE(16),SINGWV,PF(65),LMAX L4M00200
COMMON /SPOTLO/ISPOT,LOREAD,N16 L4M00210
DIMENSION TRAN(16) L4M00220
DIMENSION RADA(16),RADG(16) L4M00230
PLANCK RADIANCE FUNCTION L4M00240
FF(T,V)=1.190956E-16*(V**5)/(EXP(1.43879*V/T)-1.) L4M00250
WATT CM-2 ST-1 MICRON-1 L4M00260
***** L4M00280
PROGRAM MODIFIED LOWTRAN CALCULATES THE TRANSMITTANCE L4M00290
OF THE ATMOSPHERE FROM 830 TO 1250, 2010 TO 3330, AND L4M00300
5010 TO 39990 CM-1 (0.25 TO 2.0,3.0 TO 5.0,AND 8.0 TO 12.0 L4M00310
MICRONS) AT 20 CM-1 SPECTRAL INTERVALS ON A LINEAR WAVENUMBER L4M00320
SCALE. L4M00330
REFRACTION AND EARTH CURVATURE EFFECTS ARE EXCLUDED. ATMOSPHERE L4M00340
IS LAYERED IN ONE KM INTERVALS BETWEEN 0 AND 25 KM, 5 KM INTER- L4M00350
VALS FROM 25 TO 50 KM, A 20 KM LAYER FROM 50 TO 70 KM, A 30 KM L4M00360
LAYER FROM 70 TO 100 KM, AND ONE FROM 100 KM TO INFINITY. L4M00370
***** L4M00380
PROGRAM ACTIVATED BY SUBMISSION OF CARD SEQUENCE AS FOLLOWS L4M00390
L4M00400
CARD 1 MODEL,HAZE,ITYPE,LEN,JP,NPLT,IM,ML,IEMISS, L4M00410
RO,TBOUND,BETAEX,FORMAT(9I3,3F10.3) L4M00420
MODEL =0,METEOROLOGICAL DATA SPECIFIED L4M00430
=1,TROPICAL MODEL ATMOSPHERE L4M00440
=2,MIDLATITUDE SUMMER L4M00450
=3,MIDLATITUDE WINTER L4M00460
=4,SUBARCTIC SUMMER L4M00470
=5,SUBARCTIC WINTER L4M00480
=6,1962 US STANDARD L4M00490
=7,NEW MODEL ATMOSPHERE L4M00500
=8,ISRAELI STANDARD ATMOSPHERE (YEAR, DAYTIME) L4M00510
=9,ISRAELI STANDARD ATMOSPHERE (YEAR, NIGHTTIME) L4M00520
** AEROSOL ATTENUATION LIMITED TO 4 KM BASE HEIGHT AND 500 M THICK ** L4M00530
FOR SLANT PATHS HAZE = 1,2, OR 3 ARE THE ONLY ALLOWED VALUES. L4M00540
HAZE =0,NO AEROSOL ATTENUATION L4M00550
=1, MARITIME POLAR L4M00560
=2, MARITIME ARCTIC L4M00570
=3, CONTINENTAL POLAR L4M00580
=4, RAIN L4M00590
=5, SNOW L4M00600
=7, USER SUPPLIED EXTINCTION COEFFICIENT L4M00610
(READ ON ATM CARD - SEE CARD 3 BELOW) L4M00620
ITYPE =1,HORIZONTAL (CONSTANT PRESSURE) PATH L4M00630
=2,VERTICAL OR SLANT PATH BETWEEN 2 ALTITUDES L4M00640
=3,VERTICAL OR SLANT PATH TO SPACE L4M00650
LEN =0,NORMAL OPERATION L4M00660
=1,DOWNWARD LONG PATH L4M00670
JP =0,NORMAL OPERATION L4M00680
=1,SUPPRESS PRINT OF HORIZ AND VERTICAL PROFILES L4M00690
NPLT =0, NORMAL OPERATION L4M00700
L4M00710

```

```

      =1, IN TRANSMISSION MODE WRITE, WAVELENGTH (UM),
      H2O, CO2+, OZONE, N2 C, H2O C, MOL SCAT,
      NITRIC, SO2, HNO3, NO2, IN EMISSION MODE
      WRITE WAVELENGTH (UM) AND RADIANCE PER MICRON,
RESULTS WILL BE WRITTEN ON NPLTU (SEE COMMON BLOCK IOUNIT)
IM      =1, RADIOSONDE DATA TO BE READ INITIALLY
      =0, NORMAL OPERATION OR WHEN SUBSEQUENT CALCULATIONS
      ARE TO BE RUN WITH MODEL = 7
ML      =NUMBER OF LEVELS TO BE READ IN FOR MODEL = 7
***IM AND ML ONLY USED WHEN MODEL = 7 AND THEN ONLY ON
FIRST CALCULATIONS WHEN DATA READ IN
IEMISS  DETERMINES MODE OF EXECUTION OF PROGRAM
      =0, TRANSMITTANCE MODE
      =1, RADIANCE MODE
RO      RADIUS OF THE EARTH (KM) AT LOCATION OF CALCULATION
***DEFAULT WILL BE MIDLATITUDE VALUE OF 6371.23 KM WHEN
      MODEL = 0 OR = 7 OTHERWISE DEFAULT IS EARTH RADIUS
      FOR STANDARD MODEL ATMOSPHERE SPECIFIED BY MODEL
TBOUND  TEMPERATURE OF EARTH (DEGREES K) AT LOCATION OF CALCUL
***USED ONLY IN RADIANCE MODE FOR SLANT PATHS WHICH INTERSECT EARTH
***DEFAULT IS TEMPERATURE OF FIRST LAYER BOUNDARY TEMPERATURE
BETAEX  USER SUPPLIED EXTINCTION COEFFICIENT, INPUT ONLY
      WHEN IHAZE=7
CARD 2  H1,H2,ANGLE,RANGE,BETA,VIS,CLDHGT  FORMAT (7F10.3)
      H1      INITIAL ALTITUDE (KM)
      H2      FINAL ALTITUDE (KM)
      ANGLE    INITIAL ZENITH ANGLE (DEG)
      RANGE    PATH LENGTH (KM)
      BETA     EARTH CENTER ANGLE SUBTENDED BY H1 AND H2 (DEG)
      VIS      SEA LEVEL VISUAL RANGE (KM)
      CLDHGT   HEIGHT OF BOTTOM OF CLOUD LAYER (KM), WHEN IHAZE NE
***VIS NOT REQUIRED ON THIS CARD IF ICLMAT (EOMAIN) =1 OR
***THIS IS FIRST LOOP THROUGH LT4
***SEE MANUAL FOR MORE DETAIL
CARD 2A V1,V2,MULDV  FORMAT (2F10.3,I2)
      V1      INITIAL FREQUENCY (CM**-1)
      V2      FINAL FREQUENCY (CM**-1)
      MULDV    MULTIPLIER FOR FREQUENCY INCREMENT, WHERE THE
      INCREMENT IS A MULTIPLE OF 20 (CM**-1).
OPTIONAL CARDS FOR RESPONSE FUNTION (SET BY MR=1 IN EOMAIN)
CARD 1: NUMBER OF VALUES FOR RESPONSE FUNCTION - FORMAT (I2).
CARDS 2 - NUMBER OF VALUES: FORMAT (2(E10.4,1X))
      ONE VALUE OF WAVELENGTH (UM) AND RESPONSE FUNTION PER CARD
CARD 3  IXY  FORMAT (I3)
      IXY     =0, EXIT LOWTRAN MODULE
      =1, SELECT NEW WAVE FREQUENCY RANGE (CARD 2A)
      =2, SELECT NEW DATA SEQUENCE (CARDS 1,2,2A,3)
      =3, SELECT NEW CARD 2 AND CARD 3
      =4, SELECT NEW CARD 1 AND CARD 3
***FOR NON-STANDARD CONDITIONS SEE MANUAL
C*****
V1=V11
V2=V22
KMAX=16
ISPOT1=0
RESPFN=0.
SUMRPF=0.
SUMINT=0.
IF (ISPOT) NPLT=0
200 CONTINUE
IF (.NOT.LOREAD) GO TO 400
LOREAD=.FALSE.
READ (LOUNIT,3300) IATM,NL
NL4=4*NL
DO 299 I=1,NL4
299 READ (LOUNIT,3500) DUMMY
READ (LOUNIT,3510) PPMO2,PPMNH3,PPMNO2
READ (LOUNIT,3700)(TR(I),FW(I),FO(I),I=1,67)
READ (LOUNIT,3800)(CI(I),I=1,501)

```

READ (LOUNIT,3800)(C2(I),I=1,258)	L4M01420
READ (LOUNIT,3800)(C3(I),I=1,86)	L4M01430
READ (LOUNIT,3900)(C4(I),I=1,33)	L4M01440
READ (LOUNIT,3810)(C5(I),I=1,6)	L4M01450
READ (LOUNIT,3900)(C8(I),I=1,82)	L4M01460
READ (LOUNIT,4000)(C11(I),I=1,4)	L4M01470
READ (LOUNIT,4010)(C12(I),I=1,15)	L4M01480
READ (LOUNIT,4020)(C14(I),I=1,21)	L4M01490
READ (LOUNIT,4020)(C15(I),I=1,6)	L4M01500
READ (LOUNIT,4010)(FS(I),S1(I),S2(I),I=1,9)	L4M01510
READ (LOUNIT,4010)(FNH3(I),FH1(I),FH2(I),I=1,9)	L4M01520
READ (LOUNIT,4010)(FNO2(I),O1(I),O2(I),I=1,9)	L4M01530
REWIND LOUNIT	L4M01540
IF (ISPOT) GO TO 710	L4M01550
400 CALL CKER (V1,V2,DV,IV1,IV2,IDV,IERR,MULDV,ISPOT,TRAN(I))	L4M01560
IF (IERR.EQ.1) RETURN	L4M01570
JP=0	L4M01580
JSTOR=0	L4M01590
IP=0	L4M01600
IF (.NOT.ISPOT) GO TO 700	L4M01610
BETA=0.	L4M01620
RANGE=0.	L4M01630
RO=0.	L4M01640
IF (ITYPE.EQ.1) RANGE=ANGLE	L4M01650
IF (IXY.EQ.0) GO TO 700	L4M01660
401 GO TO (500,700,600,680),IXY	L4M01670
500 AVW=1.E+04/V1	L4M01680
ALAM=1.E+04/V2	L4M01690
SUMA=0.	L4M01700
GO TO 1100	L4M01710
600 IF (MODEL.EQ.0) GO TO 800	L4M01720
ISPOT1=1	L4M01730
GO TO 1000	L4M01740
700 CONTINUE	L4M01750
RE=6371.23	L4M01760
680 IF (.NOT.ISPOT) READ (IOIN,3300) MODEL,HAZE,ITYPE,	L4M01770
LEN,JP,NPLT,IM,ML,IEMISS,RO,TBOUND	L4M01780
C*****IEMISS=0=TRANSMISSION MODE / IEMISS=1=EMISSION MODE	L4M01790
IF ((IEMISS.EQ.1).AND.(.NOT.ISPOT)) WRITE (IOOUT,4100)	L4M01800
IF ((IEMISS.EQ.0).AND.(.NOT.ISPOT)) WRITE (IOOUT,4200)	L4M01810
IF (ISPOT) GO TO 800	L4M01820
IF (MODEL.EQ.0.OR.MODEL.EQ.7) GO TO 210	L4M01830
710 READ(LOUNIT,3300)IATM,NL	L4M01840
MSKIP=(MODEL-1)/2	L4M01850
IF(MSKIP.EQ.0) GO TO 220	L4M01860
IF (MSKIP.EQ.4) MSKIP=3	L4M01870
DO 230 J=1,MSKIP	L4M01880
DO 230 I=1,NL	L4M01890
230 READ(LOUNIT,3500) DUMMY	L4M01900
220 CONTINUE	L4M01910
C ISRAELI STD ATM READS	L4M01920
IF (MODEL.EQ.8) GO TO 270	L4M01930
IF (MODEL.EQ.9) GO TO 250	L4M01940
IF (2*(MODEL/2).EQ.MODEL) GO TO 250	L4M01950
270 DO 240 I=1,NL	L4M01960
240 READ(LOUNIT,3500)Z(I),P(I),T(I),WA(I),WH(I),WO(I)	L4M01970
GO TO 210	L4M01980
250 DO 260 I=1,NL	L4M01990
260 READ(LOUNIT,3550)Z(I),P(I),T(I),WA(I),WH(I),WO(I)	L4M02000
210 REWIND LOUNIT	L4M02010
IF (ISPOT) RETURN	L4M02020
800 M=MODEL	L4M02030
900 IF (RO.GT.0) RE=RO	L4M02040
LENTOR=LEN	L4M02050
1000 CALL ABSOR(IXY,IERR,W,V1,V2,DV,SUMA,MULDV,ANGLE,LEN,ITYPE,H1,H2,	L4M02060
MODEL,ISPOT1,RANGE,BETA,VIS,ICLMAT,IV1,IV2,IDV)	L4M02070
IF (IERR.EQ.1) TRAN(I)=1.	L4M02080
IF (IERR.EQ.1) RETURN	L4M02090
1100 CONTINUE	L4M02100
IF (.NOT.ISPOT) WRITE (IOOUT,4300)	L4M02110

	IF (.NOT.ISPOT) WRITE(IOOUT,4400) (W(I),I=1,6),W(8),W(10)	L4M02120
	IF (.NOT.ISPOT) WRITE(IOOUT,4401)	L4M02130
	IF (.NOT.ISPOT) WRITE (IOOUT,4500) W(11),W(12),W(13),W(14),	L4M02140
	* W(15),W(16)	L4M02150
1200	CONTINUE	L4M02160
C	NCLD WILL BE THE INDEX OF THE LAYER ABOVE CLDHGT	L4M02170
	DO 5 ICLD=2,6	L4M02180
	NCLD=ICLD	L4M02190
	IF (CLDHGT.LT.Z(ICLD)) GO TO 6	L4M02200
5	CONTINUE	L4M02210
6	CONTINUE	L4M02220
	IF (CLDHGT.GT.Z(6)) WRITE (IOOUT,7)	L4M02230
7	FORMAT (1H,25H**** WARNING FROM LOWTRAN,/,1X,14HCLOUD BASE IS ,	L4M02240
	1 23HLIMITED TO 4 KM MAXIMUM/)	L4M02250
	I=1	L4M02260
	L=1	L4M02270
	IV=IV1	L4M02280
	ICOUNT=0	L4M02290
	IF (N16) KWAVE=0	L4M02300
	IF (IEMISS.EQ.0) GO TO 1300	L4M02310
	RADSUM=0.0	L4M02320
	FACTOR=0.5	L4M02330
	CALL LTPATH(WLAY, WPATH, TBBY, ANGLE, LEN, ITYPE, H1, H2, MODEL)	L4M02340
	IF (.NOT.ISPOT) WRITE (IOOUT,4600)	L4M02350
	IF (.NOT.ISPOT) WRITE (IOOUT,4700)	L4M02360
C****	BEGINNING OF TRANSMITTANCE CALCULATIONS	L4M02370
1300	CONTINUE	L4M02380
	IF (N16) KWAVE=KWAVE+1	L4M02390
	SUMV=0.	L4M02400
	TLOLD=1.	L4M02410
	TSOLD=1.	L4M02420
	TX7=1.	L4M02430
	TX10=1.	L4M02440
	IKLO=1	L4M02450
	TOLD=1	L4M02460
	IF (IEMISS.EQ.0) IKMAX=IKLO	L4M02470
C	ONLY ONE LOOP FOR TRANSMISSION: LOOP OVER LAYERS FOR EMISSION	L4M02480
	DO 2300 IK=IKLO, IKMAX	L4M02490
	IF (IEMISS.EQ.0) GO TO 1500	L4M02500
C	TRANSFER CUMULATIVE ABSORBER AMOUNTS FOR TH IK TH LEVEL AND	L4M02510
C	THE K TH ABSORBER - EMISSION ONLY.	L4M02520
	DO 1400 K=1, KMAX	L4M02530
	W(K)=WPATH(IK,K)	L4M02540
1400	CONTINUE	L4M02550
1500	IJ=IK	L4M02560
	IF (ICOUNT.EQ.0) GO TO 1600	L4M02570
	IF (ICOUNT.EQ.50) GO TO 1600	L4M02580
	GO TO 1700	L4M02590
1600	ICOUNT=0	L4M02600
	IF ((IEMISS.EQ.0).AND.(.NOT.ISPOT)) WRITE (IOOUT,4800)	L4M02610
1700	DO 1800 K=1, KMAX	L4M02620
	TX(K)=1.0	L4M02630
1800	CONTINUE	L4M02640
	ICOUNT=ICOUNT+1	L4M02650
	V=FLOAT(IV)	L4M02660
	ALAM=1.E+04/V	L4M02670
	I=(IV-830)/20+1	L4M02680
	SUM4=0.	L4M02690
	SUM5=0.	L4M02700
	SUM6=0.	L4M02710
	SUM8=0.	L4M02720
	SUM11=0.	L4M02730
	CALL FREQSL(I, IV, W, TX)	L4M02740
	TX(9)=SUM4+SUM5+SUM8+SUM11+SUM6	L4M02750
	IF (TX(9).EQ.0.0) GO TO 2000	L4M02760
	IF (TX(9).LE.0.1) GO TO 1900	L4M02770
	IF (TX(9).GT.20.) GO TO 2100	L4M02780
	TX(9)=EXP(-TX(9))	L4M02790
	GO TO 2200	L4M02800
1900	TX(9)=1.0-TX(9)+0.5*TX(9)*TX(9)	L4M02810

2000	GO TO 2200	L4M02820
	TX(9)=1.0	L4M02830
	GO TO 2200	L4M02840
2100	TX(9)=0.	L4M02850
2200	TX(9)=TX(1)*TX(2)*TX(3)*TX(9)*TX(12)*TX(13)*TX(14)	L4M02860
C	AEROSOL COMPUTATIONS UNTIL LABEL 1	L4M02870
	IF (IHAZE.EQ.0.OR.IK.NE.(NCLD-1)) GO TO 1	L4M02880
C		L4M02890
C	IF SPOT IS CALLING LT4M WITH ITYPE = 3, DO NOT INCLUDE AEROSOLS	L4M02900
C		L4M02910
	IF (ISPOT.AND.ITYPE.EQ.3) GO TO 1	L4M02920
	EXT55=3.914/VIS	L4M02930
C	UPPER LIMIT OF 500 METERS VERTICAL DISTANCE FOR XSCALE	L4M02940
C	PASS HORIZONTAL DIST IF ITYPE=1, SLANT DISTANCE IF ITYPE GT 1.	L4M02950
	IF (ITYPE.EQ.1) RNG=RANGE	L4M02960
	IF (ITYPE.EQ.2.AND.(H2.GT.H1).AND.(RANGE.GT..5/COS(ANGLE*CA)))	L4M02970
1	RNG=.5/COS(ANGLE*CA)	L4M02980
	IF (ITYPE.EQ.2.AND.(H2.LT.H1).AND.	L4M02990
1	(RANGE.GT..5/COS((180.-ANGLE)*CA)))	L4M03000
2	RNG=.5/COS((180.-ANGLE)*CA)	L4M03010
	IF (ITYPE.EQ.3.AND.(RANGE.GT..5/COS(ANGLE*CA)))	L4M03020
1	RNG=.5/COS(ANGLE*CA)	L4M03030
	IF (ITYPE.EQ.3.AND.RANGE.LT..0001)	L4M03040
1	RNG=.5/COS(ANGLE*CA)	L4M03050
	ISLANT=ITYPE-1	L4M03060
C	CALL XSCALE FOR TOTAL PATH LENGTH TRANSMISSION FOR AEROSOL	L4M03070
	CALL XSCALE(AM,88.,EXT55,IX7,IERR,ISLANT,IHAZE,RNG,ANGLE)	L4M03080
	IF (IERR.EQ.1) RETURN	L4M03090
C	USER OPTIONS	L4M03100
	IF (IHAZE.EQ.7) TX7=EXP(-BETAEX*RANGE)	L4M03110
	IF (ISPOT.AND.IHAZE.EQ.8) TX7=EXP(-BE(KWAVE)*RANGE)	L4M03120
1	CONTINUE	L4M03130
	TX(9)=TX(9)*TX7	L4M03140
	IF (IV.GE.13000) TX(3)=TX(8)	L4M03150
	TNEW=TX(9)	L4M03160
	IF (IEMISS.EQ.0) GO TO 2500	L4M03170
C	COMPUTER LIMITS	L4M03180
	BBIK=0.0	L4M03190
	IF(ABS(1.43879*V/TBBY(1K)).LT.85.) BBIK=FF(TBBY(1K),V)	L4M03200
C	AEROSOL COMPUTATIONS UNTIL LABEL 2	L4M03210
	IF (IHAZE.EQ.0.OR.IK.NE.(NCLD-1)) GO TO 2	L4M03220
C	FIND AEROSOL ABSORPTION IN DIFFERENT WAVELENGTH BANDS FROM EXTN	L4M03230
	IF (ALAM.LT.2.) TX10=1.	L4M03240
	IF (ALAM.GE.3..AND.ALAM.LE.5.) TX10=TX7**2	L4M03250
	IF (ALAM.GE.8..AND.ALAM.LE.12.05) TX10=TX7**45	L4M03260
2	CONTINUE	L4M03270
	TLNEW=TX(9)*TX10/(TX(6)*TX7)	L4M03280
	TSNEW=TX7*TX(6)/TX10	L4M03290
	DTAU=ABS(TLOLD-TLNEW)	L4M03300
	IF (DTAU.LT.1.0E-5.AND.TLNEW.LT.1.0E-5) GO TO 2400	L4M03310
	SUMV=SUMV+(TOLD-TNEW)*BBIK	L4M03320
	TLOLD=TLNEW	L4M03330
	TSOLD=TSNEW	L4M03340
	TOLD=TNEW	L4M03350
2300	CONTINUE	L4M03360
2400	CONTINUE	L4M03370
	TAUG=0.	L4M03380
	IF (HMIN.LE.0.0.AND.IL.EQ.1) TAUG=TX(9)	L4M03390
	T1=T(1)	L4M03400
	IF (TBOUND.GT.0.0) T1=TBOUND	L4M03410
C	COMPUTER LIMITS	L4M03420
	BBG=0.0	L4M03430
	IF(ABS(1.43879*V/T1).LT.85.) BBG=FF(T1,V)*TAUG	L4M03440
	IF (N16) RADG(KWAVE)=BBG*1.E+04	L4M03450
	IF (N16) RADA(KWAVE)=SUMV*1.E+04	L4M03460
	IF (HMIN.LE.0) SUMV=SUMV+BBG	L4M03470
	SUMVV=SUMV	L4M03480
	IF (IV.GT.IV1) FACTOR=1.0	L4M03490
	IF (IV.GE.IV2) FACTOR=0.5	L4M03500
	SUMV=(1.0E+04/V**2)*SUMV	L4M03510

```

RADSUM=RADSUM+DV*FACTOR*SUMV
IF (.NOT. ISPOT) WRITE (100UT,4900) V,
1 ALAM,SUMV,SUMVV,RADSUM,TX(9),TX7,TX10
IF (NPLT.EQ.1) WRITE (NPLTU,98) ALAM,SUMVV
98 FORMAT (F7.4,1X,E13.5)
IF (SUMV.GE.RADMAX) VRMAX=V
IF (SUMV.GE.RADMAX) RADMAX=SUMV
IF (SUMV.LE.RADMIN) VRMIN=V
IF (SUMV.LE.RADMIN) RADMIN=SUMV
2500 AB=1.-TX(9)
IF (IV.EQ.IV1.OR.IV.GE.IV2) AB=0.5*AB
SUMA=SUMA+AB*DV
IF (IEMISS.EQ.1) GO TO 2600
IF (.NOT. ISPOT) WRITE (100UT,5000) IV,
1 ALAM,(TX(K),K=1,6),TX(11),TX(12),TX(13),TX(14),
2 SUMA,TX(9),TX7
IF (NPLT.EQ.1) WRITE (NPLTU,99) ALAM,(TX(K),K=1,6),(TX(J),J=11,14)
99 FORMAT (F7.4,10X1X,F6.4)
2600 CONTINUE
RESPFN=RESPFN(NR,ALAM)
SUMRPF=SUMRPF+RESPFN
IF (IV.GT.IV1) SUMINT=SUMINT+.5*(OLDTX9*OLDRFN+TX(9)*RESPFN)*
+ABS(1./FLOAT(IV)-1./FLOAT(IV-IDV))*1.E+04
IF (N16) TRAN(KWAVE)=TX(9)
OLDTX9=TX(9)
OLDRFN=RESPFN
IV=IV+IDV
IF (IV.GE.IV2) GO TO 2700
GO TO 1300
2700 CONTINUE
IF (NR.NE.1) SUMRPF=1.
SUMINT=SUMINT/(SUMRPF*1.E+04*ABS(1./FLOAT(IV1)-1./FLOAT(IV2)))
IF (.NOT. ISPOT) TRAN(1)=SUMINT
IF (.NOT. ISPOT.AND.NR.EQ.1) WRITE (100UT,3250) SUMINT
IF (.NOT. ISPOT.AND.NR.NE.1) WRITE (100UT,3275) SUMINT
RESPFN=0.
SUMRPF=0.
SUMINT=0.
IF ((IEMISS.EQ.1).AND.(.NOT. ISPOT)) WRITE (100UT,5100)
1 VRMIN,RADMIN,VRMAX,RADMAX
JSTOR=0
AB=1.0-SUMA/FLOAT(IV-IV1)
IF (ISPOT) RETURN
WRITE (100UT,5200) IV1,IV,SUMA,AB
IF ((IEMISS.EQ.1).AND.(.NOT. ISPOT)) WRITE (100UT,5300) RADSUM
IF (.NOT. ISPOT) READ (10IN,3300) IXY
IF (IXY.EQ.0) GO TO 3100
GO TO (2800,700,2900,680,3100),IXY
2800 CONTINUE
READ (10IN,5400) V1,V2,MULDV
CALL CKER (V1,V2,DV,IV1,IV2,IDV,IERR,MULDV,ISPOT,TRAN(1))
IF (IERR.EQ.1) RETURN
AVW=10000./V1
ALAM=10000./V2
WRITE (100UT,5500) V1,V2,DV,ALAM,AVW
SUMA=0.0
GO TO 1100
2900 IF (MODEL.EQ.0) GO TO 800
GO TO 401
3000 CONTINUE
READ (10IN,3300) MODEL,IHAZE,ITYPE,LEN,JP,NPLT,IM,
1 ML,IEMISS,RO,TBOUND,BETAEX
IF (IEMISS.EQ.1) WRITE (100UT,4100)
IF (IEMISS.EQ.0) WRITE (100UT,4200)
LENTOR=LEN
GO TO 800
3100 RETURN
3250 FORMAT (/,1X,48H WAVELENGTH AND SENSOR INTEGRATED TRANSMISSION =
+,E10.4)

```

```

3275 FORMAT (/,1X,37HWAVELENGTH INTEGRATED TRANSMISSION = ,E10.4) L4M04220
3300 FORMAT (/,9I3,3F10.3) L4M04230
3500 FORMAT (F6.1,2(E9.3,F5.1,E9.3,2E7.1)) L4M04240
3510 FORMAT(3F8.3) L4M04250
3550 FORMAT(F6.1,37X,E9.3,F5.1,E9.3,2E7.1) L4M04260
3700 FORMAT (4(F6.3,2F7.4)) L4M04270
3800 FORMAT (15F5.2) L4M04280
3810 FORMAT(15F5.3) L4M04290
3900 FORMAT (8E9.2) L4M04300
4000 FORMAT (12F6.3) L4M04310
4010 FORMAT(10F8.4) L4M04320
4020 FORMAT(10F8.3) L4M04330
4100 FORMAT (1H1,40X,36HLT4M ATMOSPHERIC TRANSMISSION MODULE,/,1X, L4M04340
1 45HPROGRAM WILL BE EXECUTED IN THE EMISSION MODE) L4M04350
4200 FORMAT (1H1,40X,36HLT4M ATMOSPHERIC TRANSMISSION MODULE,/,1X, L4M04360
1 49HPROGRAM WILL BE EXECUTED IN THE TRANSMISSION MODE) L4M04370
4300 FORMAT (/,10X,38H EQUIVALENT SEA LEVEL ABSORBER AMOUNTS L4M04380
1/,21X,56HWATER VAPOUR CO2 ETC. OZONE NITROGEN (CONT), L4M04390
2 42H H2O (CONT) MOL SCAT OZONE(U-V)/24X, L4M04400
3 7HGM CM-2,10X,2HKM,10X,6HATM CM,10X,2HKM,9X,7HGM CM-2, L4M04410
4 10X,2HKM,10X,6HATM CM) L4M04420
4400 FORMAT(/10X,10H W(1-6,8)=,7(E14.3)/,1X,10X,7H W(10)=,58X,E14.3/) L4M04430
4401 FORMAT(/23X,11HNITRIC ACID,8X,3HSO2,11X,3HNH3,11X,3HNO2/) L4M04440
4500 FORMAT(/10X,10H W(11-16)=,6(E14.3)/) L4M04450
4600 FORMAT (1H1,30X,28HRADIANCE(WATTS/CM2-STER-XXX)) L4M04460
4700 FORMAT (1H ,10X,37HFR(CM-1) WVL(MICRON) PER CM-1 L4M04470
1 10HPER MICRON,26H INTEGRAL TRANS,1X,4(1H-), L4M04480
2 11H AERO TRAN ,4(1H-),/,1X,84X,17H EXTN ABS) L4M04490
4800 FORMAT (1H1,/,1X,2X,15HFREQ WAVELENGTH,2X,3HH2O,3X,4HCO2+,4X, L4M04500
1 30HOZONE N2 C H2O C MOL S,1X, L4M04510
2 22HNITRIC SO2 HNO3,4X,16HNO2 INTEGRATED, L4M04520
3 2X,13HTOTAL AEROSOL/1X,1X,13H CM-1 MICRONS,10(3X,5HTRANS), L4M04530
4 2X,24HABSORPTION TRANS TRANS) L4M04540
4900 FORMAT (1H ,10X,F8.1,F13.6,3E13.5,F13.6,1X,F7.5,3X,F7.5) L4M04550
5000 FORMAT (1H ,16,11F8.4,F11.4,F8.4,1X,F7.5) L4M04560
5100 FORMAT (1H0,8H RADMIN ,F12.3,E12.5,/,8H RADMAX ,F12.3, L4M04570
1 E12.5) L4M04580
5200 FORMAT (1H0,26H INTEGRATED ASORPTION FROM,15,4H TO ,15, L4M04590
1 7H CM-1 =,F10.2,25H, AVERAGE TRANSMITTANCE =,F6.4) L4M04600
5300 FORMAT (1H ,22H INTEGRATED RADIANCE =,E12.5,13H WATT CM -2 ,2HSR) L4M04610
5400 FORMAT (2F10.3,12) L4M04620
5500 FORMAT (/,10X,21H FREQUENCY RANGE V1= ,F7.1,9H CM-1 TO , L4M04630
1 4HV2= ,F7.1,14H CM-1 FOR DV =,F6.1,9H CM-1 ( L4M04640
2 ,F6.2,3H - ,F5.2,10H MICRONS )) L4M04650
END L4M04660

```

```

SUBROUTINE ABSORB(IXY, IERR, W, V1, V2, DV, SUMA, MULDV, ANGLE, LEN, ITYPE, ABS00010
1 H1, H2, MODEL, ISPOT1, RANGE, BETA, VIS, ICLMAT, IV1, IV2, IDV) ABS00020
COMMON /GEOMET/PTS(15), IGESW ABS00030
COMMON /CLYMAT/TEMP, PRESS, RH1, RH1, DP1, VIS1, CLDAMT, ABS00040
1 CLDHYT, FQGRB, WNDVEL, WNDIR, IPASCT ABS00050
COMMON /CONST/PI, PI2, CA, TWOPI, TORRMB, CDEGK ABS00060
COMMON /IUNIT/IOIN, IOOUT, IPHFUN, LOUNIT, NDIRTU, NCLIMT, KSTOR, NPLTU ABS00070
COMMON /MO1/EH(16, 34), P(34), T(34), WH(34), Z(34), WAC(34), RE, M, NL ABS00080
COMMON /MO2/ WO(34), RO, TBOUND, JP, IM, ML, IP, JSTOR ABS00090
COMMON /MO9/RADMAX, RADMIN, VRMAX, VRMIN ABS00100
COMMON /EM1/HMIN, KMAX, IJ, J1, J2, JMIN, JEXTRA, NP1 ABS00110
COMMON /LQWEX/WPATH(68, 16), WLAY(34, 16), TBBY(68), TX(16), BETAEX, ABS00120
1 CLDHT, NCLD ABS00130
COMMON /SPOTLO/ISPOT, LOREAD, N16 ABS00140
COMMON /MO3/ FS(9), S1(9), S2(9), FNH3(9), FH1(9), FH2(9), FNO2(9), ABS00150
1 O1(9), O2(9), PPMO2, PPMNH3, PPMNO2 ABS00160
LOGICAL ISPOT, N16, LOREAD ABS00170
DIMENSION VH(16), WK(16), E(16) ABS00180
EH(7, 1) REPLACES HSTOR ABS00190
EH(9, 1) REPLACES HMIX ABS00200
DATA (EH(9, 1), I=1, 34) /9*0., 0.1, 0.33, 0.8, 1.2, 1.4, 1.6, 1.8, 1.9, ABS00210
1 2.0, 2.1, 2.3, 3.0, 3.7, 4.2, 5.2, 6.0, 3.8, 2.6, 0.22, 6*0.0, ABS00220
FCA=EXP(18.9766-14.9595*A-2.43882*A*A)*A ABS00230
TMPVIS=VIS ABS00240
IF (ISPOT1.EQ.1) GO TO 200 ABS00250
IF (MODEL.EQ.0) GO TO 400 ABS00260
IF (IXY.EQ.3) GO TO 100 ABS00270
IF (M.EQ.7.AND.IM.NE.0) GO TO 400 ABS00280
IF (IXY.GT.3) GO TO 1500 ABS00290
WHEN IXY=0 VIS IS READ IN MAIN ABS00300
100 IF (.NOT.ISPOT) READ (IOIN, 6200) H1, H2, ANGLE, RANGE, ABS00310
1 BETA, VIS, CLDHT ABS00320
IF (IGESW.NE.1) GO TO 111 ABS00330
H1=PTS(3) ABS00340
H2=PTS(6) ABS00350
RANGE=SQRT((PTS(1)-PTS(4))**2+(PTS(2)-PTS(5))**2+ ABS00360
+ (PTS(3)-PTS(6))**2) ABS00370
111 CONTINUE ABS00380
IF (IXY.EQ.0) VIS=TMPVIS ABS00390
200 X1=RE+H1 ABS00400
X2=RE+H2 ABS00410
IF (ITYPE.EQ.3) GO TO 1000 ABS00420
IF (ITYPE.EQ.1) GO TO 1500 ABS00430
IF (RANGE.EQ.0.) GO TO 1200 ABS00440
IF (.NOT.ISPOT) WRITE (IOOUT, 6300) H1, H2, ANGLE, RANGE, ABS00450
1 BETA, VIS ABS00460
IF (H2.EQ.0.AND.ANGLE.NE.0) GO TO 300 ABS00470
ANGLE=ACOS(0.5*((H2-H1)*(1.+X2/X1)/RANGE-RANGE/X1))/CA ABS00480
GO TO 1400 ABS00490
300 X2=SQRT((X1/RANGE+RANGE/X1+2.0*COS(ANGLE*CA))*X1* ABS00500
1 RANGE) ABS00510
H2=X2-RE ABS00520
GO TO 1400 ABS00530
400 CONTINUE ABS00540
IF (ML.LE.0) ML=1 ABS00550
DO 900 K=1, ML ABS00560
CLIMATE OPTION - SEE COMMON /CLYMAT/ ABS00570
IF (M.EQ.0.AND..NOT.ISPOT) READ (IOIN, 6400) H1, P(1), ABS00580
1 TMP, DP, RH, WH(K), WO(K), VIS, RANGE ABS00590
IF (IGESW.NE.1) GO TO 444 ABS00600
H1=PTS(3) ABS00610
H2=PTS(6) ABS00620
RANGE=SQRT((PTS(1)-PTS(4))**2+(PTS(2)-PTS(5))**2+ ABS00630
+ (PTS(3)-PTS(6))**2) ABS00640
444 CONTINUE ABS00650
IF (ICLMAT.NE.1) GO TO 500 ABS00660
TMP=TEMP ABS00670
P(1)=PRESS ABS00680
DP=DP1 ABS00690
RH=RH1 ABS00700

```



```

      WH(K)=AH1
      VIS=VIS1
500  IF (IXY.EQ.0.AND.ICLMAT.NE.1) VIS=TMPVIS
      IF (M.GT.0).AND.(.NOT.ISPOT) READ (IOIN,6400) Z(K),
1    P(K),TMP,DP,RH,WH(K),WO(K)
      IF (M.EQ.0).AND.(.NOT.ISPOT) WRITE (IOOUT,6500) H1,
1    P(1),TMP,DP,RH,WH(K),WO(K),VIS,RANGE
      IF (M.EQ.0) Z(K)=H1
      J=IFIX(Z(K)+1.0E-6)+1
      IF (Z(K).GE.25.) J=(Z(K)-25.)/5.0+26.
      IF (Z(K).GE.50.0) J=(Z(K)-50.)/20.+31.
      IF (Z(K).GE.70.) J=(Z(K)-70.)/30.+32.
      IF (J.GT.33) J=33
      FAC=Z(K)-FLOAT(J-1)
      IF (J.LT.26) GO TO 600
      FAC=(Z(K)-5.0*FLOAT(J-26)-25.)/5.
      IF (J.GE.31) FAC=(Z(K)-50.)/20.
      IF (J.GE.32) FAC=(Z(K)-70.)/30.
      IF (FAC.GT.1.0) FAC=1.0
600  L=J+1
      T(K)=TMP+CDEGK
      TT=CDEGK/T(K)
      IF (RH.LE.0.0) TT=CDEGK/(CDEGK+DP)
      IF (WH(K).LE.0.0) WH(K)=F(TT)
      IF (RH.GT.0.0) WH(K)=0.01*RH*WH(K)
      EH(7,K)=0.0
      IF (EH(9,J).LE.0.) GO TO 700
      EH(7,K)=EH(9,J)*(EH(9,L)/EH(9,J))*FAC
700  CONTINUE
      IF (MODEL.EQ.0) GO TO 1500
      IF (K.EQ.1).AND.(.NOT.ISPOT) WRITE (IOOUT,6600)
      IF (.NOT.ISPOT) WRITE (IOOUT,6400) Z(K),P(K),TMP,DP,
1    RH,WH(K),WO(K)
900  CONTINUE
      IM=0
      NL=ML
C     NOTE THAT Z(1) MAY NOT CORRESPOND TO THE VALUES GIVEN FOR STANDARD
C     MODEL ATMOSPHERES
      IF (IXY.GE.3) GO TO 1500
      GO TO 100
1000 IF (RANGE.GT.0.0) GO TO 1100
      GO TO 1500
1100 ITYPE=2
      BETA=ACOS(0.5*(RANGE*RANGE/(X1*X2)-X2/X1-X1/X2))/CA
1200 IF (BETA.EQ.0.) GO TO 1300
      BET=CA*BETA
      X2=RE+H2
      ANGLE=ATAN(X2*SIN(BET)/(X2*COS(BET)-X1))/CA
      IF (ANGLE.LT.0.) ANGLE=ANGLE+PI
      RANGE=X2*SIN(BET)/SIN(ANGLE*CA)
      BET=BETA
      GO TO 1500
1300 RANGE=(X2/X1)**2-(SIN(ANGLE*CA))**2
      IF (RANGE.GE.0.0) RANGE=X1*(SQRT(RANGE)-ABS(COS(ANGLE*CA)))
1400 IF (ANGLE.NE.0.,OR,ANGLE.NE.180.) BET=ASIN(RANGE*SIN(ANGLE*CA)/X2)
      IF (ANGLE.LT.0.) ANGLE=ANGLE+PI
      IF (RANGE.LT.0.0) RANGE=-RANGE
      BET=BET/CA
      IF (.NOT.ISPOT) WRITE (IOOUT,6300) H1,H2,ANGLE,RANGE.
1    BET,VIS
1500 CONTINUE
      DO 1600 I=1,NL
      DO 1600 J=1,KMAX
1600 WLAY(I,J)=0.
      SUMA=0.
C     WHEN IXY=0 V1,V2,MULDV ARE READ IN EOSAEL.MAIN
      IF (IXY.EQ.1.OR,IXY.EQ.2).AND.(.NOT.ISPOT)
1    READ (IOIN,6250) V1,V2,MULDV
      IF (IXY.EQ.1.OR,IXY.EQ.2).AND.(.NOT.ISPOT)
+ CALL CKER (V1,V2,DV,IV1,IV2,IOV,IERR,MULDV,ISFJT,TMPVIS)

```

```

ABS00710
ABS00720
ABS00730
ABS00740
ABS00750
ABS00760
ABS00770
ABS00780
ABS00790
ABS00800
ABS00810
ABS00820
ABS00830
ABS00840
ABS00850
ABS00860
ABS00870
ABS00880
ABS00890
ABS00900
ABS00910
ABS00920
ABS00930
ABS00940
ABS00950
ABS00960
ABS00970
ABS00980
ABS00990
ABS01000
ABS01010
ABS01020
ABS01030
ABS01040
ABS01050
ABS01060
ABS01070
ABS01080
ABS01090
ABS01100
ABS01110
ABS01120
ABS01130
ABS01140
ABS01150
ABS01160
ABS01170
ABS01180
ABS01190
ABS01200
ABS01210
ABS01220
ABS01230
ABS01240
ABS01250
ABS01260
ABS01270
ABS01280
ABS01290
ABS01300
ABS01310
ABS01320
ABS01330
ABS01340
ABS01350
ABS01360
ABS01370
ABS01380
ABS01390
ABS01400

```

IF (IERR.EQ.1) RETURN	ABS01410
IF ((ITYPE.EQ.1).AND.(.NOT.ISPOT)) WRITE (IOOUT,6700)	ABS01420
1 H1,RANGE	ABS01430
IF ((ITYPE.EQ.2).AND.(.NOT.ISPOT)) WRITE (IOOUT,6800)	ABS01440
1 H1,H2,ANGLE	ABS01450
IF ((ITYPE.EQ.3).AND.(.NOT.ISPOT)) WRITE (IOOUT,6900)	ABS01460
1 H1,ANGLE	ABS01470
IF (MODEL.EQ.0) M=7	ABS01480
IF ((M.EQ.1).AND.(.NOT.ISPOT)) WRITE (IOOUT,7200) M	ABS01490
IF ((M.EQ.2).AND.(.NOT.ISPOT)) WRITE (IOOUT,7300) M	ABS01500
IF ((M.EQ.3).AND.(.NOT.ISPOT)) WRITE (IOOUT,7400) M	ABS01510
IF ((M.EQ.4).AND.(.NOT.ISPOT)) WRITE (IOOUT,7500) M	ABS01520
IF ((M.EQ.5).AND.(.NOT.ISPOT)) WRITE (IOOUT,7600) M	ABS01530
IF ((M.EQ.6).AND.(.NOT.ISPOT)) WRITE (IOOUT,7700) M	ABS01540
IF ((M.EQ.8).AND.(.NOT.ISPOT)) WRITE (IOOUT,7800) M	ABS01550
IF ((M.EQ.9).AND.(.NOT.ISPOT)) WRITE (IOOUT,7900) M	ABS01560
AVW=10000./V1	ABS01570
ALAM=10000./V2	ABS01580
RADMIN=1.0E+38	ABS01590
RADMAX=0.	ABS01600
VRMIN=0.	ABS01610
VRMAX=0.	ABS01620
IF (.NOT.ISPOT) WRITE (IOOUT,8000) V1,V2,DV,ALAM,AVW	ABS01630
AVW=0.5E-4*(V1+V2)	ABS01640
AVW=AVW*AVW	ABS01650
IF ((JP.EQ.0).AND.(.NOT.ISPOT)) WRITE (IOOUT,8100)	ABS01660
IF (ITYPE.EQ.1) GO TO 2100	ABS01670
DO 1800 K=1,KMAX	ABS01680
VH(K)=0.0	ABS01690
1800 CONTINUE	ABS01700
BETA=0.0	ABS01710
SR=0.0	ABS01720
IP=0	ABS01730
C**** NOW DEFINE CONSTANT PRESSURE PATH QUANTITIES EH(1-8)	ABS01740
Y=CA*ANGLE	ABS01750
SPHI=SIN(Y)	ABS01760
R1=(RE+H1)*SPHI	ABS01770
IF (H1.GT.Z(NL)) GO TO 1900	ABS01780
GO TO 2100	ABS01790
1900 X=(RE+Z(NL))/(RE+H1)	ABS01800
IF (SPHI.GT.X) GO TO 2000	ABS01810
H1=Z(NL)	ABS01820
J1=NL	ABS01830
SPHI=SPHI/X	ABS01840
ANGLE=180.0-ASIN(SPHI)/CA	ABS01850
R1=(RE+H1)*SPHI	ABS01860
GO TO 2100	ABS01870
2000 HMIN=R1-RE	ABS01880
IF (.NOT.ISPOT) WRITE (IOOUT,8200) HMIN	ABS01890
GO TO 6000	ABS01900
2100 DO 2400 I=1,NL	ABS01910
PS=P(I)/1013.0	ABS01920
TS=CDEGK/T(I)	ABS01930
X=PS*TS	ABS01940
C--- COMPUTE MASS DENSITY (G M-3) FROM IDEAL GAS LAW ---	ABS01950
C--- 1292.02 = DENSITY OF STANDARD COMPOSITION AIR AT STP ---	ABS01960
WA(I) = 1292.02*X	ABS01970
PT=PS*SQR(TS)	ABS01980
D=0.1*WH(I)	ABS01990
EH(1,I)=D*PT**0.9	ABS02000
EH(2,I)=X*PT**0.75	ABS02010
EH(4,I)=0.8*PT*X	ABS02020
PPW=4.56E-5*D*CDEGK/TS	ABS02030
TS1=(296.0/CDEGK)*TS	ABS02040
EH(5,I)=D*PPW*EXP(6.08*(TS1-1.0))+0.002*D*(PS-PPW)	ABS02050
EH(10,I)=D*(PPW+0.12*(PS-PPW))*EXP(4.56*(TS1-1.0))	ABS02060
EH(6,I)=X	ABS02070
EH(8,I)=46.6667*W0(I)	ABS02080
EH(3,I)=EH(8,I)*PT**0.4	ABS02090
C EH(11,I)=HN03 ABSORBER AMOUNT (ATM-CM)/KM	ABS02100

```

      EH(11,1)=PS*TS*EH(9,1)*1.0E-04
      IF (MODEL.EQ.0.OR.MODEL.EQ.7) EH(11,1)=PS*TS*EH(7,1)*1.0E-04
C*****EH(12,1)=S02  ABSORBER AMOUNT (ATM-CM)/KM
      EH(12,1)=0.772E-04*PPMSO2*WA(I)*PS**0.07122*TS**0.06159
      EH(13,1)=0.772E-04*PPMNH3*WA(I)*PS**0.52125*TS**(-0.60438)
      EH(14,1)=0.772E-04*PPMNO2*WA(I)*PS**0.18066*TS**0.20911
C***K=15 FOR ASL 3.3 - 4.3 MICRON H2O CONTINUUM
      EH(15,1)=PPW*0
C***K=16 FOR 4.6 - 4.8 MICRON H2O CONTINUUM
      EH(16,1)=PPW*(PPW+3.0E-03*(PS-PPW))
      IF (I.EQ.NL) GO TO 2300
      IF (MODEL.EQ.0.AND.I.GE.1) GO TO 3600
      T2=T(I+1)
      W2=W(I+1)
      PPW=4.56E-6*W2*T2
2300 IF (H1.GE.Z(I)) J1=I
      IF ((JP.EQ.0).AND.(.NOT.ISPOT))
1WRITE (100UT,8300)I,Z(I),(EH(K,I),K=1,6),EH(8,I),(EH(K,I),K=10,14)
2400 CONTINUE
      X1=H1
      CALL POINT (H1,N,NP1,TX)
      J1=N
      DO 2500 K=1,KMAX
2500 E(K)=TX(K)
      JEXTRA=0
      JMIN=0
C**ITYPE=1 MEANS HORIZONTAL PATH *****
      IF (ITYPE.EQ.1) GO TO 3600
      IF (ITYPE.EQ.3) H2=Z(NL)
C** ANGLE GREATER THAN 90 DEGREES MEANS DOWNWARD TRAJECTORY *****
      IF (ANGLE.GT.90.0) GO TO 3800
C** IF THE PATH IS NOT HORIZONTAL OR DOWNWARD THEN IT IS UPWARD TRAJECTORY
2600 IF (ANGLE.GT.90.0.AND.NP1.GT.0) J1=J1+1
      J2=NL
      IF (ITYPE.EQ.3) GO TO 2700
      CALL POINT (H2,N,NP,TX)
      J2=N
      IF (NP.GT.0) J2=J2-1
2700 DO 2800 K=1,KMAX
      IF (K.EQ.9.OR.K.EQ.7) GO TO 2800
      EH(K,J1)=E(K)
      IF (ITYPE.EQ.3) GO TO 2800
      EH(K,J2+1)=TX(K)
2800 CONTINUE
C***** NOW DEFINE VERTICAL PATH QUANTITIES VH(1-8)
      IF ((JP.EQ.0).AND.(.NOT.ISPOT)) WRITE (100UT,8400)
2900 W(K)=0
      DO 3500 I=J1,J2
      X1=Z(I)
      IF (I.LT.NL) X2=Z(I+1)
      IF (I.EQ.NL) X2=Z(I)
      IF (I.EQ.J1) X1=H1
      IF (I.EQ.J2) X2=H2
      DZ=X2-X1
      IF (I.EQ.NL) DZ=Z(I)-Z(I-1)
      DS=DZ
C***** UPWARD TRAJECTORY
      RX=(RE+X1)/(RE+X2)
      THETA=ASIN(SPHI)/CA
      PHI=ASIN(SPHI*RX)/CA
      BET=THETA-PHI
      SALP=RX*SPHI
      IF (SPHI.GT.1.E-10) DS=(RE+X2)*SIN(BET*CA)/SPHI
      BETA=BETA+BET
      PSI=BETA+PHI-ANGLE
      PHI=180.-PHI
      SR=SR+DS
      JEXTRA=0
      DO 3400 K=1,KMAX

```

```

ABS02110
ABS02120
ABS02130
ABS02140
ABS02150
ABS02160
ABS02170
ABS02180
ABS02190
ABS02200
ABS02210
ABS02220
ABS02230
ABS02240
ABS02250
ABS02260
ABS02270
ABS02280
ABS02290
ABS02300
ABS02310
ABS02320
ABS02330
ABS02340
ABS02350
ABS02360
ABS02370
ABS02380
ABS02390
ABS02400
ABS02410
ABS02420
ABS02430
ABS02440
ABS02450
ABS02460
ABS02470
ABS02480
ABS02490
ABS02500
ABS02510
ABS02520
ABS02530
ABS02540
ABS02550
ABS02560
ABS02570
ABS02580
ABS02590
ABS02600
ABS02610
ABS02620
ABS02630
ABS02640
ABS02650
ABS02660
ABS02670
ABS02680
ABS02690
ABS02700
ABS02710
ABS02720
ABS02730
ABS02740
ABS02750
ABS02760
ABS02770
ABS02780
ABS02790
ABS02800

```

IF (K.EQ.7.OR.K.EQ.9) GO TO 3100	ABS02810
EV=DS*EH(K,I)	ABS02820
IF (I.EQ.NL) GO TO 3000	ABS02830
IF (EH(K,I).EQ.0.0.OR.EH(K,I+1).EQ.0.0) GO TO 3100	ABS02840
IF (EH(K,I).EQ.EH(K,I+1)) GO TO 3200	ABS02850
EV=DS*(EH(K,I)-EH(K,I+1))/ALOG(EH(K,I)/EH(K,I+1))	ABS02860
GO TO 3200	ABS02870
3000 IF (EH(K,I).EQ.0.0) GO TO 3100	ABS02880
IF (EH(K,I-1).EQ.0.0) GO TO 3100	ABS02890
IF (EH(K,I).EQ.EH(K,I-1)) GO TO 3200	ABS02900
EV=EV/ALOG(EH(K,I-1)/EH(K,I))	ABS02910
GO TO 3200	ABS02920
3100 EV=0.	ABS02930
3200 VH(K)=VH(K)+EV	ABS02940
IF (I.EQ.JSTOR) GO TO 3300	ABS02950
W(LAY(I,K)=EV+W(K)	ABS02960
W(K)=0.	ABS02970
GO TO 3400	ABS02980
3300 W(K)=EV	ABS02990
IF (J1.NE.J2) GO TO 3400	ABS03000
W(LAY(J2+1,K)=W(K)	ABS03010
W(K)=0.	ABS03020
JEXTRA=1	ABS03030
3400 CONTINUE	ABS03040
IF ((JP.EQ.0).AND.(.NOT.ISPOT)) WRITE (IOOUT,8500) I,	ABS03050
X1,(VH(L),L=1,6),VH(8),PSI,PHI,BETA,THETA,SR	ABS03060
IF (I.GE.NL) GO TO 3500	ABS03070
SPHI=SPHI+RX	ABS03080
IF (SALP.GE.1.) SPHI=SALP	ABS03090
3500 CONTINUE	ABS03100
GO TO 5800	ABS03110
C**** HORIZONTAL PATH	ABS03120
3600 DO 3700 K=1,KMAX	ABS03130
IF (K.EQ.7.OR.K.EQ.9) GO TO 3700	ABS03140
W(K)=RANGE*EH(K,1)	ABS03150
IF (MODEL.GT.0) W(K)=RANGE*TX(K)	ABS03160
VH(K)=W(K)	ABS03170
3700 CONTINUE	ABS03180
GO TO 6100	ABS03190
3800 CONTINUE	ABS03200
C**** DOWNWARD TRAJECTORY	ABS03210
K2=0	ABS03220
IF (NP1.EQ.1) J1=J1-1	ABS03230
IF (J1.LE.0) J1=1	ABS03240
J2=J1+1	ABS03250
J=J1+1	ABS03260
IF (H2.GT.Z(J1+1).OR.H1.EQ.H2) GO TO 4000	ABS03270
IF (NP1.EQ.1.AND.H2.GE.Z(J1+1)) GO TO 4000	ABS03280
CALL POINT (H2,N,NP2,TX)	ABS03290
DO 3900 K=1,KMAX	ABS03300
3900 W(K)=TX(K)	ABS03310
IF (H2.LT.H1) H=H2	ABS03320
J2=N	ABS03330
4000 A0=(RE+H1)*SPHI	ABS03340
DO 4100 I=1,J1	ABS03350
HMIN=A0-RE	ABS03360
JMIN=1	ABS03370
IF (HMIN.LE.Z(I+1)) GO TO 4200	ABS03380
4100 CONTINUE	ABS03390
4200 X=HMIN	ABS03400
IF (HMIN.LE.0) GO TO 4400	ABS03410
CALL POINT (X,N,NP,TX)	ABS03420
JMIN=N	ABS03430
HMIN=A0-RE	ABS03440
IF (ABS(X-HMIN).GT.0.0001) GO TO 4200	ABS03450
IF (H2.GE.H1) J2=N	ABS03460
IF (H2.GE.H1.OR.H2.LT.HMIN) H=HMIN	ABS03470
IF (.NOT.ISPOT) WRITE (IOOUT,8600) HMIN	ABS03480
IF (H2.LT.HMIN) J2=N	ABS03490
IF ((H2.LT.HMIN).AND.(.NOT.ISPOT)) WRITE (IOOUT,8700) HMIN	ABS03500

```

GO TO 4500
4400 IF (.NOT.ISPOT) WRITE (100UT,8600) HMIN
      IF (H2.LT.H1) GO TO 4500
      IF ((ITYPE.EQ.3.OR.H2.GE.H1).AND.(.NOT.ISPOT))
1      WRITE (100UT,8800)
      ITYPE=2
      JMIN=0
      J2=1
      H2=0.0
      H=0.0
C**** NOW DEFINE VERTICAL PATH QUANTITIES VH(1-8)
4500 IF ((JP.EQ.0).AND.(.NOT.ISPOT)) WRITE (100UT,8400)
      JSTOR=J-1
      DO 5100 I=1,NL
      J=J-1
      IF (I.NE.1) X1=Z(J+1)
      X2=Z(J)
      IF (J.EQ.J2.AND.K2.EQ.0) X2=H
      IF (J.EQ.JMIN.AND.K2.EQ.1) X2=HMIN
      HM=(RE+X1)*SPHI-RE
      IF (HM.GT.Z(J).AND.HM.GT.X2) X2=HM
      RX=(RE+X1)/(RE+X2)
      DS=X1-X2
      ALP=90.0
      THET=ASIN(SPHI)/CA
      SALP=RX*SPHI
      IF (ABS(2-HM).GT.1.0E-5) ALP=ASIN(SALP)/CA
      BET=ALP-THET
      IF (SPHI.GT.1.0E-10) DS=(RE+X2)*SIN(BET*CA)/SPHI
      THETA=180.0-THET
      BETA=BETA+BET
      PSI=BETA-ALP-ANGLE+180.0
      SR=SR+DS
      DO 5000 K=1,KMAX
      IF (K.EQ.7.OR.K.EQ.9) GO TO 5000
      AJ=EH(K,J)
      BJ=EH(K,J+1)
      IF (J.EQ.J1) BJ=E(K)
      IF (J.EQ.J2.AND.H2.LT.H1.AND.H2.GT.0.0) AJ=W(K)
      IF (J.EQ.JMIN.AND.H2.GE.H1) AJ=TX(K)
      IF (J.EQ.JMIN.AND.ABS(H2-HM).LT.1.0E-5) AJ=TX(K)
      IF (K2.EQ.0) GO TO 4600
      IF (J.EQ.J2) BJ=W(K)
      IF (J.EQ.JMIN) AJ=TX(K)
4600 IF (AJ.EQ.0.0.OR.BJ.EQ.0.0) GO TO 4800
      IF (AJ.EQ.BJ) GO TO 4700
      EV=DS*(AJ-BJ)/ALOG(AJ/BJ)
      GO TO 4900
4700 EV=DS*AJ
      GO TO 4900
4800 EV=0.0
4900 VH(K)=VH(K)+EV
5000 WLAY(J,K)=EV
      IF ((JP.EQ.0).AND.(.NOT.ISPOT)) WRITE (100UT,8500) J,
1      X1,(VH(L),L=1,6),VH(8),PSI,ALP,BETA,THETA,SR
      IF (J.EQ.J2.AND.H2.GE.H1) GO TO 5600
      IF (J.EQ.JMIN.AND.K2.EQ.1) GO TO 5400
      SPHI=SALP
      IF (J.EQ.J2.AND.K2.EQ.0) GO TO 5200
5100 CONTINUE
5200 IF (HMIN.LE.0) GO TO 5800
      IF ((LEN.EQ.0).AND.(.NOT.ISPOT)) WRITE (100UT,8900)
      IF ((LEN.EQ.1).AND.(.NOT.ISPOT)) WRITE (100UT,9000)
      IF (LEN.EQ.0) GO TO 5800
      K2=1
      X1=X2
      IF (ABS(X1-HMIN).LE.0.001) GO TO 5800
      H=HMIN
      J=J2+1
      IF (NP2.EQ.1) J=J-1

```

```

ABS03510
ABS03520
ABS03530
ABS03540
ABS03550
ABS03560
ABS03570
ABS03580
ABS03590
ABS03600
ABS03610
ABS03620
ABS03630
ABS03640
ABS03650
ABS03660
ABS03670
ABS03680
ABS03690
ABS03700
ABS03710
ABS03720
ABS03730
ABS03740
ABS03750
ABS03760
ABS03770
ABS03780
ABS03790
ABS03800
ABS03810
ABS03820
ABS03830
ABS03840
ABS03850
ABS03860
ABS03870
ABS03880
ABS03890
ABS03900
ABS03910
ABS03920
ABS03930
ABS03940
ABS03950
ABS03960
ABS03970
ABS03980
ABS03990
ABS04000
ABS04010
ABS04020
ABS04030
ABS04040
ABS04050
ABS04060
ABS04070
ABS04080
ABS04090
ABS04100
ABS04110
ABS04120
ABS04130
ABS04140
ABS04150
ABS04160
ABS04170
ABS04180
ABS04190
ABS04200

```

B=BETA	ABS04210
PH=180.0- ASIN(SPHI)/CA	ABS04220
TS=SR	ABS04230
PS=PSI	ABS04240
DO 5300 K=1,KMAX	ABS04250
5300 E(K)=VH(K)	ABS04260
GO TO 4500	ABS04270
5400 BETA=2.*BETA-B	ABS04280
PSI=2.*PSI-PS	ABS04290
SR=2.*SR-TS	ABS04300
C LONG PATH TAKEN	ABS04310
PHI=PH	ABS04320
DO 5500 K=1,KMAX	ABS04330
5500 VH(K)=2.*VH(K)-E(K)	ABS04340
GO TO 5800	ABS04350
5600 DO 5700 K=1,KMAX	ABS04360
5700 VH(K)=2.0*VH(K)	ABS04370
BETA=2.0*BETA	ABS04380
SR=2.0*SR	ABS04390
IF (H2.EQ.H1) GO TO 5800	ABS04400
SPHI=SIN(ANGLE*CA)	ABS04410
GO TO 2600	ABS04420
5800 CONTINUE	ABS04430
DO 5900 K=1,KMAX	ABS04440
W(K)=VH(K)	ABS04450
5900 CONTINUE	ABS04460
6000 CONTINUE	ABS04470
6100 RETURN	ABS04480
C	ABS04490
6200 FORMAT (7F10.3)	ABS04500
6250 FORMAT (2F10.3,12)	ABS04510
6300 FORMAT (1H0.9X,4H H1=,F7.3,6HKM,H2=,F7.3,9HKM,ANGLE=,	ABS04520
1 F8.4,13HGEOM. RANGE =,F7.2,8HKM,BETA=,F8.5,	ABS04530
2 5H,VIS=,F6.1)	ABS04540
6400 FORMAT (3F10.3,2F5.1,2E10.3,2F10.3)	ABS04550
6500 FORMAT (10X,26HINPUT METEOROLOGICAL DATA:/10X,2HZ=,	ABS04560
1 F7.2,7H KM, P=,F7.2,6H MB,T=,F5.1,8H C, DEW ,	ABS04570
2 7HPT,TEMP,F5.1,17H C, REL HUMIDITY=,F5.1,	ABS04580
3 16H %, H2O DENSITY=,1PE9.2,7H GM M-3/10X,	ABS04590
4 15H OZONE DENSITY=,E9.2,20H GM-3, VISUAL RANGE=	ABS04600
5 2OPF6.1,10H KM,RANGE=,F10.3,4H KM)	ABS04610
6600 FORMAT (24H MODEL ATMOSPHERE NO. 7,4X,6HZ (KM),3X,	ABS04620
1 6HP (MB),4X,30HT (C) DEW PT %RH H2O(GM.M-3) ,	ABS04630
2 19H03(GM.M-3) NO. DEN.)	ABS04640
6700 FORMAT (/10X,28H HORIZONTAL PATH, ALTITUDE =,F7.3,	ABS04650
1 11H KM,RANGE =,F7.3,3H KM)	ABS04660
6800 FORMAT (/10X,37H SLANT PATH BETWEEN ALTITUDES H1 AND ,	ABS04670
1 13HH2 WHERE H1 =,F7.3,8H KM H2 =,F7.3,	ABS04680
2 18H KM,ZENITH ANGLE =,F7.3,8H DEGREES)	ABS04690
6900 FORMAT (/10X,39H SLANT PATH TO SPACE FROM ALTITUDE H1 =	ABS04700
1 ,F7.3,19H KM, ZENITH ANGLE =,F7.3,	ABS04710
2 8H DEGREES)	ABS04720
7200 FORMAT (/20X,18H MODEL ATMOSPHERE ,11,11H = TROPICAL)	ABS04730
7300 FORMAT (/20X,18H MODEL ATMOSPHERE ,11,	ABS04740
1 21H = MIDLATITUDE SUMMER)	ABS04750
7400 FORMAT (/20X,18H MODEL ATMOSPHERE ,11,	ABS04760
1 21H = MIDLATITUDE WINTER)	ABS04770
7500 FORMAT (/20X,18H MODEL ATMOSPHERE ,11,14H = SUB-ARCTIC ,	ABS04780
1 7HEUMMER)	ABS04790
7600 FORMAT (/20X,18H MODEL ATMOSPHERE ,11,14H = SUB-ARCTIC ,	ABS04800
1 7HWINTER)	ABS04810
7700 FORMAT (/20X,18H MODEL ATMOSPHERE ,11,11H = 1962 US ,	ABS04820
1 10HSTANDARD)	ABS04830
7800 FORMAT(/20X,18H MODEL ATMOSPHERE ,11,20H = ISRAELI STANDARD ,	ABS04840
1 16H(YEAR, DAYTIME))	ABS04850
7900 FORMAT(/20X,18H MODEL ATMOSPHERE ,11,20H = ISRAELI STANDARD ,	ABS04860
1 18H(YEAR, NIGHTTIME))	ABS04870
8000 FORMAT (/10X,21H FREQUENCY RANGE V1= ,F7.1,9H CM-1 TO ,	ABS04880
1 4HV2= ,F7.1,14H CM-1 FOR DV =,F6.1,9H CM-1 (ABS04890
2 ,F6.2,3H - ,F5.2,10H MICRONS)	ABS04900

8100	FORMAT	(1H1,///10X,20H HORIZONTAL PROFILES/)	ABS04910
8200	FORMAT	(38H TRAJECTORY MISSES EARTHS ATMOSPHERE, ,1X,/ 31HCLOSEST DISTANCE OF APPROACH IS,F10.2,1X,/ 1X,18HEND OF CALCULATION)	ABS04920 ABS04930 ABS04940
8300	FORMAT	(1X,14,F6.1,12(E9.3))	ABS04950
8400	FORMAT	(1H1,///10X,21H VERTICAL PROFILES ,53X,3HPSI, 6X,3HPhi,6X,4HBETA,3X,13HTheta RANGE)	ABS04960 ABS04970
8500	FORMAT	(13,F6.1,7E10.3,4F9.4,F6.1)	ABS04980
8600	FORMAT	(8H HMIN = ,F10.3)	ABS04990
8700	FORMAT	(40H H2 WAS SET LESS THAN HMIN AND HAS BEEN , 34HRESET EQUAL TO HMIN I.E. H2 = ,F10.3)	ABS05000 ABS05010
8800	FORMAT	(41H PATH INTERSECTS EARTH - PATH CHANGED TO , 23HTYPE 2 WITH H2 = 0.0 KM)	ABS05020 ABS05030
8900	FORMAT	(36H CHOICE OF TWO PATHS FOR THIS CASE - , 42HSHORTEST PATH TAKEN. FOR LONGER PATH SET , 6HLEN=1.)	ABS05040 ABS05050 ABS05060
9000	FORMAT	(44H CHOICE OF TWO PATHS FOR THIS CASE -LONGEST , 41HPATH TAKEN. FOR SHORT PATH SET LEN = 0)	ABS05070 ABS05080 ABS05090
	1	END	

	SUBROUTINE CKER (V1,V2,DV,IV1,IV2,IDV,IERR,MULDV,ISPOT,TRAN)	CKR00010
	LOGICAL ISPOT	CKR00020
	COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU	CKR00030
	IERR=0	CKR00040
	IF (MULDV.LE.0) MULDV=1	CKR00050
C	CHECK FOR PROPER WINDOW REGION	CKR00060
	IF (V2.LE.830.0) GO TO 1090	CKR00070
	IF (V1.GE.1250.0.AND.V2.LE.2010.0) GO TO 1090	CKR00080
	IF (V1.GE.3330.0.AND.V2.LT.5010.0) GO TO 1090	CKR00090
	IF (V1.GE.39990.0) GO TO 1090	CKR00100
	GO TO 1100	CKR00110
1090	WRITE(IOOUT,3190)	CKR00120
	TRAN=1.	CKR00130
	IERR=1	CKR00140
	RETURN	CKR00150
C	CHECK FOR PROPER INTEGER VALUES	CKR00160
1100	IV1=20*FIX((V1-10.0)/20.0)+10	CKR00170
	IV2=20*FIX((V2-10.0)/20.0+0.99)+10	CKR00180
	IF (IV1.LT.830) IV1=830	CKR00190
	IF (IV1.LT.1250.AND.IV2.GT.1250) IV2=1250	CKR00200
	IF (IV1.LT.2010.AND.IV2.GT.2010) IV1=2010	CKR00210
	IF (IV1.LT.3330.AND.IV2.GT.3330) IV2=3330	CKR00220
	IF (IV1.LT.5010.AND.IV2.GT.5010) IV1=5010	CKR00230
	IF (IV1.LT.39990.AND.IV2.GT.39990) IV2=39990	CKR00240
	V1=FLOAT(IV1)	CKR00250
	V2=FLOAT(IV2)	CKR00260
	DV=20.*FLOAT(MULDV)	CKR00270
	IWVCK=0	CKR00280
	IDV=FIX(DV)	CKR00290
C	WHEN CALLED FROM SPOT CHECK FOR MORE THAN 15 DIVISIONS	CKR00300
C	(16 WAVENUMBER VALUES) WHICH IS ARRAY SIZE.	CKR00310
C	CHECK HERE FOR ROUNDING PROBLEMS ON V1, V2 CAUSING	CKR00320
C	TOO SMALL AN INCREMENT	CKR00330
	IF (ISPOT.AND.(FLOAT(IV2-IV1)/15..GT.(FLOAT(IDV)+.001))) IWVCK=1	CKR00340
	IF (IWVCK.NE.1) GO TO 91	CKR00350
	CKDV=FLOAT(IV2-IV1)/(15.*20.)	CKR00360
	MULDV=FIX(CKDV)	CKR00370
	IF (FLOAT(MULDV)/CKDV.LT..99) MULDV=MULDV+1	CKR00380
	DVHOLD=20.*FLOAT(MULDV)	CKR00390
	WRITE (IOOUT,93) DV,DVHOLD	CKR00400
	DV=DVHOLD	CKR00410
	IF (DV.LT.20.) DV=20.	CKR00420
	IDV=FIX(DV)	CKR00430
93	FORMAT (1X,'DIVISION LIMITS CHANGED FROM ',F10.3,	CKR00440
	+ ' TO ',F10.3)	CKR00450
91	CONTINUE	CKR00460
	RETURN	CKR00470
3190	FORMAT (6X,'*****FREQUENCY IS OUTSIDE OF THE WINDOW*****'/	CKR00480
1	6X,'*****TOTAL TRANSMITTANCE IS 1.0000*****')	CKR00490
	END	CKR00500

	SUBROUTINE FREQSL(I,IV,W,IX)	FRE00010
	COMMON /M05/ C1(501),C2(258),C3(86),C4(33),C5(6),C5DUM(9),C8(82),	FRE00020
	1 C11(4),C12(15),C14(21),C15(6)	FRE00030
	COMMON /M07/ TR(67),FW(67),FO(67)	FRE00040
	COMMON /M08/SUM4,SUM5,SUM8,SUM11,SUM6	FRE00050
	DIMENSION W(16),IX(16)	FRE00060
	IF (I.EQ.1) GO TO 10	FRE00070
	IF (I.GE.2.AND.I.LE.3) GO TO 11	FRE00080
	IF (I.GE.4.AND.I.LE.5) GO TO 12	FRE00090
	IF (I.GE.6.AND.I.LE.12) GO TO 13	FRE00100
	IF (I.GE.13.AND.I.LE.21) GO TO 15	FRE00110
	IF (I.EQ.22) GO TO 16	FRE00120
	IF ((I.GE.23.AND.I.LE.59).OR.(I.GE.127.AND.I.LE.209)) RETURN	FRE00130
	IF ((I.GE.60.AND.I.LE.63)) GO TO 14	FRE00140
	IF (I.GE.64.AND.I.LE.76) GO TO 18	FRE00150
	IF ((I.GE.77.AND.I.LE.81).OR.(I.GE.87.AND.I.LE.96)) GO TO 18	FRE00160
	IF (I.GE.82.AND.I.LE.86) GO TO 30	FRE00170
	IF ((I.GE.97.AND.I.LE.101).OR.(I.GE.105.AND.I.LE.109)) GO TO 14	FRE00180
	IF (I.GE.102.AND.I.LE.104) GO TO 9	FRE00190
	IF (I.GE.110.AND.I.LE.112) GO TO 21	FRE00200
	IF (I.GE.113.AND.I.LE.123) GO TO 22	FRE00210
	IF ((I.GE.124.AND.I.LE.126).OR.(I.GE.210.AND.I.LE.363)) GO TO 23	FRE00220
	IF ((I.GE.364.AND.I.LE.419).OR.(I.GE.454.AND.I.LE.599)) GO TO 24	FRE00230
	IF ((I.GE.420.AND.I.LE.453).OR.(I.GE.600.AND.I.LE.606).OR.(I.GE	FRE00240
	1,1160.AND.I.LE.1334)) GO TO 35	FRE00250
	IF (I.GE.607.AND.I.LE.609) GO TO 25	FRE00260
	IF (I.GE.610.AND.I.LE.621) GO TO 26	FRE00270
	IF ((I.GE.622.AND.I.LE.629).OR.(I.GE.686.AND.I.LE.1159).OR.(I.GE	FRE00280
	1,1335)) GO TO 27	FRE00290
	IF (I.GE.630.AND.I.LE.685) GO TO 28	FRE00300
4	CALL H2OVAR(I,W,C1,IX)	FRE00310
	GO TO 40	FRE00320
6	CALL OZONE(I,W,C3,IX)	FRE00330
7	CALL UNIMIX(I,W,C2,IX)	FRE00340
	GO TO 4	FRE00350
8	CALL NO2(I,W,C15,IX)	FRE00360
	GO TO 6	FRE00370
9	CALL H2O410(I,IV,W,C5,IX,SUM5)	FRE00380
	GO TO 8	FRE00390
10	CALL NH3(I,W,C14,IX)	FRE00400
	GO TO 9	FRE00410
11	CALL NITRIC(I,W,C11,SUM11,IX)	FRE00420
	GO TO 10	FRE00430
12	CALL NITRIC(I,W,C11,SUM11,IX)	FRE00440
13	CALL NH3(I,W,C14,IX)	FRE00450
14	CALL H2O410(I,IV,W,C5,IX,SUM5)	FRE00460
	GO TO 6	FRE00470
15	CALL SO2(I,W,C12,IX)	FRE00480
	GO TO 13	FRE00490
16	CALL SO2(I,W,C12,IX)	FRE00500
	GO TO 14	FRE00510
18	CALL H2O410(I,IV,W,C5,IX,SUM5)	FRE00520
19	CALL NITRO(I,W,C4,IX,SUM4)	FRE00530
	GO TO 6	FRE00540
20	CALL SO2(I,W,C12,IX)	FRE00550
	GO TO 10	FRE00560
21	CALL MOLSC(I,IV,W,IX,SUM6)	FRE00570
	GO TO 14	FRE00580
22	CALL MOLSC(I,IV,W,IX,SUM6)	FRE00590
	GO TO 6	FRE00600
23	CALL MOLSC(I,IV,W,IX,SUM6)	FRE00610
	GO TO 7	FRE00620
24	CALL MOLSC(I,IV,W,IX,SUM6)	FRE00630
	GO TO 4	FRE00640
25	CALL UNIMIX(I,W,C2,IX)	FRE00650
35	CALL MOLSC(I,IV,W,IX,SUM6)	FRE00660
	GO TO 40	FRE00670
26	CALL UVOZNE(I,W,C8,IX,SUM8)	FRE00680
	GO TO 25	FRE00690
27	CALL UVOZNE(I,W,C8,IX,SUM8)	FRE00700

CALL MOLSCT(I,V,W,TX,SUM6)
GO TO 40
28 CALL H2OVAP(I,W,C1,TX)
GO TO 27
30 CALL H2O410(I,I,V,W,C5,TX,SUM5)
GO TO 20
40 RETURN
END

FRE00710
FRE00720
FRE00730
FRE00740
FRE00750
FRE00760
FRE00770
FRE00780

```

SUBROUTINE H2OVAP(I,W,C1,TX)
DIMENSION C1(501),TX(1),W(1)
C*****TRANSMITTANCE FOR WATER VAPOR*****
IF (W(1).LT.1.0E-20) GO TO 500
IF (I.LE.22) I1=I
IF (I.GE.60.AND.I.LE.126) I1=I-37
IF (I.GE.210.AND.I.LE.419) I1=I-120
IF (I.GE.454.AND.I.LE.599) I1=I-154
IF (I.GE.630) I1=I-184
WS1=ALOG10(W(1))+C1(I1)
TX(1)=EXP(-10**(-1.14619+0.55013*WS1))
500 RETURN
END

```

```

H2V00010
H2V00020
H2V00030
H2V00040
H2V00050
H2V00060
H2V00070
H2V00080
H2V00090
H2V00100
H2V00110
H2V00120
H2V00130

```

SUBROUTINE H20410(I,IV,W,C5,TX,SUM5)	H2F00010
C***WATER VAPOR CONTINUUM, 3-5 AND 8-12 MICRON REGIONS	H2F00020
DIMENSION C5(6),TX(5),W(16)	H2F00030
V=FLOAT(IV)	H2F00040
IF(I.GT.109) GO TO 100	H2F00050
IF(I.GT.22) GO TO 200	H2F00060
C***1.LE.I.LE.22:CALCULATE OPTICAL DEPTH DUE TO 8-12 MICRON CONTINUUM	H2F00070
TX(5)=(4.18+5578.0*EXP(-7.87E-03*V))*W(5)	H2F00080
GO TO 300	H2F00090
200 CONTINUE	H2F00100
IF(I.LT.63) GO TO 100	H2F00110
IF(I.GT.68) GO TO 400	H2F00120
C***63.LE.I.LE.68:CALCULATE OPTICAL DEPTH DUE TO 4.6-4.8 MICRON	H2F00130
C***CONTINUUM MODEL FROM BEN-SHALOM ET AL.1980,SPIE,VOL.253,261.	H2F00140
II=I-62	H2F00150
TX(5)=C5(II)*577.6*W(16)	H2F00160
GO TO 300	H2F00170
400 CONTINUE	H2F00180
IF(I.LT.77) GO TO 100	H2F00190
C***77.LE.I.LE.109:CALCULATE OPTICAL DEPTH DUE TO 3.3-4.3 MICRON	H2F00200
C***CONTINUUM MODEL FROM WATKINS ET AL.1979,APPL.OPT.,VOL.18,1149.	H2F00210
V=V*1.0E-03	H2F00220
V2=V*V	H2F00230
V3=V2*V	H2F00240
CBURCH=46.4745-48.0898*V+16.3988*V2-1.83217*V3	H2F00250
CASL=-370.082+508.137*V-225.822*V2+32.7744*V3	H2F00260
TX(5)=CBURCH*W(10)+CASL*W(15)	H2F00270
GO TO 300	H2F00280
100 TX(5)=0.0	H2F00290
300 SUM5=TX(5)	H2F00300
IF(TX(5).LT.1.0E-05) GO TO 500	H2F00310
IF(TX(5).GT.20.0) GO TO 600	H2F00320
TX(5)=EXP(-TX(5))	H2F00330
RETURN	H2F00340
500 TX(5)=1.0	H2F00350
RETURN	H2F00360
600 TX(5)=0.0	H2F00370
RETURN	H2F00380
END	H2F00390

```

SUBROUTINE LTPATH(WLAY, WPATH, TBBY, ANGLE, LEN, ITYPE, H1, H2, MODEL) LTP00010
COMMON /MO1/EH(16,34),P(34),T(34),WH(34),Z(34),WA(34),RE,M,NL LTP00020
DIMENSION WLAY(34,16),TBBY(68),WPATH(68,16) LTP00030
COMMON /EM1/HMIN,KMAX,IJ,J1,J2,JMIN,JEXTRA,NP1 LTP00040
COMMON /EM2/W(16),E(16),IL,IKMAX,LENTOR,NLL LTP00050
COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLINT,KSTOR,NPLOTUL LTP00060
COMMON /SPOTLO/ISPOT,LOREAD,N16 LTP00070
LOGICAL ISPOT,N16,LOREAD LTP00080
555 IL=0 LTP00090
IF (ITYPE.EQ.1) GO TO 1000 LTP00100
IF (ITYPE.EQ.2.AND.H1.EQ.H2) J2=J1 LTP00110
IF (H2.GT.H1.AND.ANGLE.GT.90..AND.NP1.EQ.1) J1=J1-1 LTP00120
IF (JEXTRA.EQ.1) J2=J2+1 LTP00130
IF ((ITYPE.EQ.2).AND.(H1.GT.H2).AND.(LENTOR.EQ.1)) J2= LTP00140
J2-1 LTP00150
1 IF (ITYPE.EQ.3) J2=NLL LTP00160
IF (.NOT.ISPOT) WRITE (IOOUT,1200) LTP00170
DO 100 IK=1,68 LTP00180
TBBY(IK)=0. LTP00190
DO 100 K=1,KMAX LTP00200
WPATH(IK,K)=0. LTP00210
100 CONTINUE LTP00220
LEN=0 LTP00230
NLL=NLL-1 LTP00240
IL=J1+1 LTP00250
IJ=IL+NLL LTP00260
DO 200 K=1,KMAX LTP00270
E(K)=0. LTP00280
200 CONTINUE LTP00290
IF (ANGLE.GT.90.0) GO TO 300 LTP00300
LEN=1. LTP00310
IL=J1-1 LTP00320
HMIN=1.0E-6 LTP00330
IJ=NLL LTP00340
300 CONTINUE LTP00350
DO 800 IK=1,68 LTP00360
IF (LEN.EQ.0) IL=IL-1 LTP00370
IF (LEN.EQ.1) IL=IL+1 LTP00380
IJ=IJ-1 LTP00390
IF (IL.EQ.0) GO TO 800 LTP00400
DO 400 K=1,KMAX LTP00410
W(K)=E(K)+WLAY(IL,K) LTP00420
WPATH(IK,K)=W(K) LTP00430
400 CONTINUE LTP00440
IF (IL.LE.0.OR.IL.GE.NLL) GO TO 500 LTP00450
TBAR=(T(IL)+T(IL+1))*0.5 LTP00460
C JEXTRA IS 1 ONLY WHEN PROGRAM NEVER LEAVES ONE LAYER LTP00470
IF (JEXTRA.EQ.1) TBAR=(T(J1)+T(J1+1))*0.5 LTP00480
500 CONTINUE LTP00490
TBBY(IK)=TBAR LTP00500
DO 600 K=1,KMAX LTP00510
E(K)=W(K) LTP00520
600 CONTINUE LTP00530
IF (ANGLE.LE.90.0.AND.IL.EQ.NLL) GO TO 900 LTP00540
IF (ITYPE.EQ.3.AND.ANGLE.LE.90.0) GO TO 700 LTP00550
IF (ITYPE.EQ.3.AND.LEN.EQ.1.AND.IL.EQ.J2) GO TO 900 LTP00560
IF (ITYPE.EQ.2.AND.LENTOR.EQ.0.AND.IL.EQ.J2) GO TO 900 LTP00570
IF (IL.EQ.JMIN.AND.HMIN.GT.0) LEN=1 LTP00580
IF (IL.EQ.1.AND.HMIN.LE.0.0) GO TO 900 LTP00590
IF (LEN.EQ.0) GO TO 700 LTP00600
IF (IL.EQ.JMIN.AND.IJ.EQ.IL+NLL) IL=IL-1 LTP00610
IF (ITYPE.EQ.2.AND.IL.EQ.J2) GO TO 900 LTP00620
700 CONTINUE LTP00630
IF (.NOT.ISPOT) WRITE (IOOUT,1300) IK,(WPATH(IK,K),K= LTP00640
1,6),WPATH(IK,8),(WPATH(IK,K),K=10,14),TBBY(IK) LTP00650
800 CONTINUE LTP00660
IKMAX=68 LTP00670
LEN=LENTOR LTP00680
RETURN LTP00690
900 CONTINUE LTP00700

```

IF (.NOT.ISPOT) WRITE (IOOUT,1300) IK,(WPATH(IK,K),K=	LTP00710
1 1,6),WPATH(IK,8),WPATH(IK,K),K=10,14),TBBY(IK)	LTP00720
IKMAX=IK	LTP00730
LEN=LENTOR	LTP00740
RETURN	LTP00750
1000 DO 1100 K=1,KMAX	LTP00760
WPATH(1,K)=W(K)	LTP00770
1100 CONTINUE	LTP00780
IF (MODEL.EQ.0) J1=1	LTP00790
J2=J1	LTP00800
TBBY(1)=T(J1)	LTP00810
IKMAX=1	LTP00820
IK=1	LTP00830
IF (.NOT.ISPOT) WRITE (IOOUT,1200)	LTP00840
IF (.NOT.ISPOT) WRITE (IOOUT,1300) IK,(WPATH(IK,K),K=	LTP00850
1 1,6),WPATH(IK,8),WPATH(IK,K),K=10,14),TBBY(IK)	LTP00860
HMIN=1.0E-6	LTP00870
RETURN	LTP00880
C 1200 FORMAT (//,20X,37H CUMULATIVE ABSORBER AMOUNTS FOR THE ,	LTP00890
1 16HATMOSPHERIC PATH,// 8X,3HH2O,5X,4HC02+,6X,	LTP00900
2 2HO3,7X,2HN2,6X,5HH2O C,4X,5HMOL S,4X,5HO3 UV,	LTP00910
3 4X,5HH2O C,5X,4HHNO3,6X,3HSO2,6X,3HHH3,6X,3HN02,	LTP00920
4 5X,4HTAVE)	LTP00930
1300 FORMAT (15,12E9.3,F10.3)	LTP00940
END	LTP00950
	LTP00960

SUBROUTINE MOLSCT(IV,W,TX,SUM6)	MOL00010
C*****TRANSMITTANCE FOR MOLECULAR SCATTERING *****	MOL00020
C C6 EXPRESSION MODIFIED AS PER SHETTLER ET AL 1980,APPL.OPT.,VOL.19,	MOL00030
C 2873	MOL00040
DIMENSION TX(6),W(6)	MOL00050
V=FLOAT(IV)	MOL00060
C6=(V**4)/(9.67578E+18-1.11836E+09*V**2)	MOL00070
TX(6)=C6*W(6)	MOL00080
SUM6=TX(6)	MOL00090
IF (TX(6).EQ.0.0) GO TO 200	MOL00100
IF (TX(6).LE.0.1) GO TO 100	MOL00110
IF (TX(6).GT.20.) GO TO 300	MOL00120
TX(6)=EXP(-TX(6))	MOL00130
GO TO 400	MOL00140
100 TX(6)=1.0-TX(6)+0.5*TX(6)*TX(6)	MOL00150
GO TO 400	MOL00160
200 TX(6)=1.0	MOL00170
GO TO 400	MOL00180
300 TX(6)=0.0	MOL00190
400 RETURN	MOL00200
END	MOL00210

	SUBROUTINE NH3(I,W,C13,TX)	NHA00010
	COMMON /M03/ FS(9),S1(9),S2(9),FNH3(9),FH1(9),FH2(9),FN02(9),	NHA00020
	1 O1(9),O2(9),PPMS02,PPMNH3,PPMN02	NHA00030
	DIMENSION C13(21),TX(13),W(13)	NHA00040
C	*****	NHA00050
C	THIS SUBROUTINE CALCULATES THE TRANSMITTANCE BY NH3 (PPM READ IN	NHA00060
C	THE MAIN PROGRAM).	NHA00070
C	*****	NHA00080
	I1=1	NHA00090
	IF (W(13).LT.1.0E-20) GO TO 3	NHA00100
	WS13=ALOG10(W(13))+C13(I1)	NHA00110
	DO 1 J=1,9	NHA00120
	IF (WS13-FNH3(J)) 2,2,1	NHA00130
1	CONTINUE	NHA00140
2	TX(13)=EXP(-10**((FH1(J)+FH2(J))*WS13))	NHA00150
3	RETURN	NHA00160
	END	NHA00170
		NHA00180
		NHA00190

SUBROUTINE NITRIC(I,W,C11,SUM11,TX)	NITC0010
DIMENSION C11(4),TX(11),W(11)	NITC0020
C***** TRANSMITTANCE FOR NITRIC ACID*****	NITC0030
HABS=0.	NITC0040
IF (I.LT.2.OR.I.GT.46) GO TO 100	NITC0050
IF (I.GT.5.AND.I.LT.23) GO TO 100	NITC0060
I1=I-1	NITC0070
HABS=C11(I1)	NITC0080
100 CONTINUE	NITC0090
TX(11)=HABS*W(11)	NITC0100
SUM11=TX(11)	NITC0110
IF (TX(11).EQ.0.0) GO TO 300	NITC0120
IF (TX(11).LE.0.1) GO TO 200	NITC0130
IF (TX(11).GT.20.) GO TO 400	NITC0140
TX(11)=EXP(-TX(11))	NITC0150
GO TO 500	NITC0160
200 TX(11)=1.0-TX(11)+0.5*TX(11)*TX(11)	NITC0170
GO TO 500	NITC0180
300 TX(11)=1.0	NITC0190
GO TO 500	NITC0200
400 TX(11)=0.0	NITC0210
500 RETURN	NITC0220
END	NITC0230

SUBROUTINE NITRO(I,W,C4,TX,SUM4)	NITR0010
DIMENSION C4(33),TX(4),W(4)	NITR0020
C*****TRANSMITTANCE FOR NITROGEN CONTINUUM*****	NITR0030
IF (I.LT.64) GO TO 200	NITR0040
I1=I-63	NITR0050
C TEMP FIX FOLLOWS	NITR0060
C IF (I1.GT.10) GO TO 300	NITR0070
TX(4)=C4(I1)*W(4)	NITR0080
SUM4=TX(4)	NITR0090
IF (TX(4).EQ.0.0) GO TO 200	NITR0100
IF (TX(4).LE.0.1) GO TO 100	NITR0110
IF (TX(4).GT.20.) GO TO 300	NITR0120
TX(4)=EXP(-TX(4))	NITR0130
GO TO 400	NITR0140
100 TX(4)=1.0-TX(4)+0.5*TX(4)*TX(4)	NITR0150
GO TO 400	NITR0160
200 TX(4)=1.0	NITR0170
GO TO 400	NITR0180
300 TX(4)=0.0	NITR0190
400 RETURN	NITR0200
END	NITR0210

	SUBROUTINE NO2(I,W,C14,IX)	NO2X0010
	COMMON /NO3/ FS(9),S1(9),S2(9),FHH3(9),FH1(9),FH2(9),FNO2(9),	NO2X0020
	1 O1(9),O2(9),PPMSO2,PPMNH3,PPMNO2	NO2X0030
	DIMENSION C14(6),TX(14),W(14)	NO2X0040
C*****		NO2X0050
C	THIS SUBROUTINE CALCULATES THE TRANSMITTANCE BY NO2 (PPM READ IN	NO2X0060
C	THE MAIN PROGRAM).	NO2X0070
C		NO2X0080
C*****		NO2X0090
	IF (I.GE.102.AND.I.LE.104) I1=I-98	NO2X0100
	IF (I.LE.3) I1=1	NO2X0110
	IF (W(14).LT.1.0E-20) GO TO 3	NO2X0120
	WS14=ALOG10(W(14))+C14(I1)	NO2X0130
	DO 1 J=1,9	NO2X0140
	IF (WS14-FNO2(J)) 2,2,1	NO2X0150
1	CONTINUE	NO2X0160
2	TX(14)=EXP(-10**((O1(J)+O2(J))*WS14))	NO2X0170
3	RETURN	NO2X0180
	END	NO2X0190
		NO2X0200

SUBROUTINE OZONE(I,W,C3,TX)	OZN00010
DIMENSION C3(86),TX(3),W(3)	OZN00020
C*****TRANSMITTANCE FOR OZONE*****	OZN00030
IF (W(3).LT.1.0E-20) GO TO 500	OZN00040
IF (I.LE.22) I1=I	OZN00050
IF (I.GE.60) I1=I-37	OZN00060
WS3=ALOG10(W(3))+C3(I1)	OZN00070
TX(3)=1/(1+EXP(-3.08019+2.11127*WS3))	OZN00080
500 RETURN	OZN00090
END	OZN00100

```

SUBROUTINE POINT (X,N,NP,TX)
COMMON /M01/EH(16,34),P(34),T(34),WH(34),Z(34),WA(34),RE,M,NL
COMMON /EM1/HMIN,KMAX,IJ,J1,J2,JMIN,JEXTRA,NP1
COMMON /IQUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLINT,KSTOR,NPLOTU
COMMON /SPOTLO/ISPOT,LOREAD,N16
DIMENSION TX(16)
LOGICAL ISPOT,N16,LOREAD
C*****
C***** SUBROUTINE POINT INTERPOLATES EXPONENTIALLY TO
C***** DETERMINE THE EQUIVALENT ABSORBER AMOUNTS AT THAT ALTITUDE.
C*****
C***** X IS THE HEIGHT IN QUESTION
C***** N IS THE LEVEL INTEGER CORRESPONDING TO X OR THE LEVEL BELOW X
C***** NP = 1 IF X COINCIDES WITH MODEL ATMOSPHERE LEVEL, IF NOT NP = 0
C***** TX(1-8) ARE ABSORBER AMOUNTS PER KM AT HEIGHT X
C*****
555 N=NL
NP=0
IF (X.LT.0.0) X=0.
IF (X.GT.Z(NL)) GO TO 400
DO 100 I=1,NL
N=I
IF (X-Z(I)) 200,400,100
100 CONTINUE
200 J2=N
N=N-1
FAC=(X-Z(N))/(Z(J2)-Z(N))
DO 300 K=1,KMAX
IF (K.EQ.9.OR.K.EQ.7) GO TO 300
TX(K)=0.0
IF (EH(K,N).EQ.0.0) GO TO 300
IF (EH(K,N).GT.1000.0) GO TO 300
TX(K)=EH(K,N)*(EH(K,J2)/EH(K,N))*FAC
300 CONTINUE
GO TO 700
400 NP=1
DO 500 K=1,KMAX
500 TX(K)=EH(K,N)
700 RETURN
END

```

POI00010
 POI00020
 POI00030
 POI00040
 POI00050
 POI00060
 POI00070
 POI00080
 POI00090
 POI00100
 POI00110
 POI00120
 POI00130
 POI00140
 POI00150
 POI00160
 POI00170
 POI00180
 POI00190
 POI00200
 POI00210
 POI00220
 POI00230
 POI00240
 POI00250
 POI00260
 POI00270
 POI00280
 POI00290
 POI00300
 POI00310
 POI00320
 POI00330
 POI00340
 POI00350
 POI00360
 POI00370
 POI00380
 POI00390
 POI00400

```

C      FUNCTION RESFN (NR,WAVE)
C      THIS FUNCTION WILL READ IN UP TO 20 VALUES OF A RESPONSE FUNCTION
C      IF THE RESF CARD IS READ IN EOMAIN. ONLY ONE RESPONSE FUNCTION
C      PER RUN IS ALLOWED. THIS FUNCTION WILL ALSO DO A LINEAR INTERPOLARE
C      OVER WAVELENGTH. IF RESF CARD IS NOT READ A VALUE OF 1 IS RETURNED
C      TO THE CALLING PROGRAM WHEN THIS FUNCTION IS REFERENCED.
COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUR
DIMENSION WAVELN(20),RESPFN(20)
DATA WAVELN,RESPFN,ICOUNT,NBR /20*0.,20*0.,0,1/
IF (NR.NE.1) GO TO 6
ICOUNT=ICOUNT+1
IF (ICOUNT.GT.1) GO TO 2
READ (IOIN,100) NBR
IF (NBR.GT.20) WRITE (IOOUT,102)
IF (NBR.GT.20) STOP
WRITE (IOOUT,103)
DO 3 I=1,NBR
  READ (IOIN,101) WAVELN(I),RESPFN(I)
  WRITE (IOOUT,104) WAVELN(I),RESPFN(I)
2  IF (WAVE.LT.(WAVELN(1)-.0001).OR.WAVE.GT.(WAVELN(NBR)+.0001))
+ GO TO 6
DO 4 I=1,NBR
  K=I
4  IF (WAVE.GE.WAVELN(I)) GO TO 5
5  IF (WAVE/WAVELN(K).GE..99.AND.WAVE/WAVELN(K).LE.1.01) GO TO 7
  IF (K.EQ.NBR) GO TO 8
  RESFN=(WAVE-WAVELN(K))*(RESPFN(K+1)-RESPFN(K))/
1 (RESPFN(K+1)-RESPFN(K))+RESPFN(K)
  RETURN
7  RESFN=RESPFN(K)
  RETURN
8  RESFN=RESPFN(NBR)
  RETURN
6  RESFN=1.
  RETURN
100  FORMAT (I2)
101  FORMAT (2(E10.4,1X))
102  FORMAT (1H,51H THE NUMBER OF VALUES FOR THE RESPONSE FUNCTION IS G
+58H REATER THAN THE DIMENSIONS LIMITS OF WAVELN( ) AND RESPFN( ),
+/,1X,19H PROGRAM TERMINATED.)
103  FORMAT (1H,20X,23H INPUT RESPONSE FUNCTION,/,1X,15X,10HWAVELENGTH,
+5X,10HR FUNCTION)
104  FORMAT (1H,15X,2(E10.4,1X))
END
RES00010
RES00020
RES00030
RES00040
RES00050
RES00060
RES00070
RES00080
RES00090
RES00100
RES00110
RES00120
RES00130
RES00140
RES00150
RES00160
RES00170
RES00180
RES00190
RES00200
RES00210
RES00220
RES00230
RES00240
RES00250
RES00260
RES00270
RES00280
RES00290
RES00300
RES00310
RES00320
RES00330
RES00340
RES00350
RES00360
RES00370
RES00380
RES00390
RES00400
RES00410
RES00420
RES00430
RES00440

```

	SUBROUTINE S02(I,W,C12,TX)	S02X0010
	COMMON /M03/ FS(9),S1(9),S2(9),FHH3(9),FH1(9),FH2(9),FN02(9),	S02X0020
	1 O1(9),O2(9),PPMS02,PPMNH3,PPMN02	S02X0030
	DIMENSION C12(15),TX(12),W(12)	S02X0040
C	*****	S02X0050
C	THIS SUBROUTINE CALCULATES THE TRANSMITTANCE BY SO2 (PPM READ IN	S02X0060
C	THE MAIN PROGRAM).	S02X0070
C	*****	S02X0080
C	IF (W(12).LT.1.0E-20) GO TO 5	S02X0090
	IF (I.GE.13.AND.I.LE.22) I1=I-12	S02X0100
	IF (I.GE.82) I1=I-71	S02X0110
	WS12=ALOG10(W(12))+C12(I1)	S02X0120
	DO 1 J=1,9	S02X0130
	IF(WS12-FS(J)) 2,2,1	S02X0140
1	CONTINUE	S02X0150
2	TX(12)=EXP(-10**((S1(J)+S2(J)*WS12))	S02X0160
3	RETURN	S02X0170
	END	S02X0180
		S02X0190
		S02X0200

SUBROUTINE UNIMIX(I,W,C2,TX)	UNI00010
DIMENSION C2(258),TX(2),W(2)	UNI00020
C*****TRANSMITTANCE FOR UNIFORMLY MIXED GASES*****	UNI00030
IF (W(2).LT.1.0E-20) GO TO 500	UNI00040
IF (I.LE.22) I1=I	UNI00050
IF (I.GE.60.AND.I.LE.126) I1=I-37	UNI00060
IF (I.GE.210.AND.I.LE.363) I1=I-120	UNI00070
IF (I.GE.607) I1=I-363	UNI00080
WS2=ALOG10(W(2))+C2(I1)	UNI00090
TX(2)=EXP(-10**(-1.14619+0.55013*WS2))	UNI00100
500 RETURN	UNI00110
END	UNI00120

SUBROUTINE UVOZNE(I,W,C8,TX,SUM8)	UVZ00010
DIMENSION C8(82),TX(8),W(8)	UVZ00020
C*****+ TRANSMITTANCE FOR UV OZONE *****	UVZ00030
AI=I	UVZ00040
IF(I.LE.1159) GO TO 90	UVZ00050
IF(I.GE.1335) GO TO 100	UVZ00060
90 XX=10.0	UVZ00070
XI=(AI-610.0)/XX+1.0	UVZ00080
L1=1	UVZ00090
L2=53	UVZ00100
GO TO 200	UVZ00110
100 XX=25.0	UVZ00120
XI=(AI-1335.0)/XX+57.0	UVZ00130
L1=57	UVZ00140
L2=102	UVZ00150
200 DO 300 N=L1,L2	UVZ00160
XD=XI-FLOAT(N)	UVZ00170
IF(XD) 500,400,300	UVZ00180
300 CONTINUE	UVZ00190
400 TX(8)=W(8)*C8(N)	UVZ00200
GO TO 600	UVZ00210
500 TX(8)=C8(N)+XD*(C8(N)-C8(N-1))	UVZ00220
TX(8)=W(8)*TX(8)	UVZ00230
600 SUM8=TX(8)	UVZ00240
IF(TX(8).EQ.0.0) GO TO 800	UVZ00250
IF(TX(8).LE.0.1) GO TO 700	UVZ00260
IF(TX(8).GT.20.0) GO TO 900	UVZ00270
TX(8)=EXP(-TX(8))	UVZ00280
GO TO 1000	UVZ00290
700 TX(8)=1.0-TX(8)+0.5*TX(8)*TX(8)	UVZ00300
GO TO 1000	UVZ00310
800 TX(8)=1.0	UVZ00320
GO TO 1000	UVZ00330
900 TX(8)=0.0	UVZ00340
1000 RETURN	UVZ00350
END	UVZ00360

```

SUBROUTINE SPOT(WAVN1,WAVN2,VIS,NR,IERR,MULDV)
*****
INPUT: EXCLUDING THE OPTIONAL RESPONSE FUNCTION CARDS,
THERE IS A MAXIMUM OF 7 CARDS TO EXECUTE THIS MODULE.
THE CARDS MAY BE INSERTED IN ANY ORDER WITH THE EXCEPTION OF
THE LAST CARD WHICH SIGNIFIES THAT EXECUTION IS TO BEGIN.
THE CARDS ARE INPUT WITH FORMAT (A4,6X,7E10.4)
EACH CARD BEGINS WITH A 4 LETTER IDENTIFIER IN COL 1 - 4
FOLLOWED BY AS MANY (REAL) FIELDS AS NEEDED, 10 COL PER
FIELD BEGINNING IN COL 11.
THE CARDS ARE NOT ORDER DEPENDENT.
CARD 1
ENVR  ISORC, ITARG, IHAZE, MODEL, NLAM
      ISORC = 0 SUNLIGHT ONLY
            1 MOONLIGHT ONLY
            2 EMISSION ONLY
            3 SUNLIGHT AND EMISSION
            4 MOONLIGHT AND EMISSION
      ITARG = 0 BACKGROUND ONLY
            1 GROUND REFLECTANCE/EMISSION
            2 TARGET REFLECTANCE/EMISSION
** AEROSOL ATTENUATION LIMITED TO 4 KM BASE HEIGHT AND 500 M THICK **
FOR SLANT PATHS IHAZE = 1,2, OR 3 ARE THE ONLY ALLOWED VALUES.
      IHAZE = 0, NO AEROSOL ATTENUATION
            = 1, MARITIME POLAR
            = 2, MARITIME ARCTIC
            = 3, CONTINENTAL POLAR
            = 4, RAIN
            = 5, SNOW
            = 7, USER SUPPLIED EXTINCTION COEFFICIENT
              (READ ON ATM CARD - SEE CARD 3 BELOW)
            = 8, EXTINCTION COEFFICIENT WILL BE READ FROM
              PHASE FUNCTION DATA FILE
      MODEL = 1 TROPICAL MODEL ATMOSPHERE
            2 MIDLATITUDE SUMMER
            3 MIDLATITUDE WINTER
            4 SUBARCTIC SUMMER
            5 SUBARCTIC WINTER
            6 1962 US STANDARD
            8 ISRAELI STANDARD (YEAR, DAYTIME)
            9 ISRAELI STANDARD (YEAR, NIGHTTIME)
      NLAM  = 0 NO AEROSOL ATTENUATION
            = 1 READ PHASE FUNCTION DATA SET - ALSO SEE
              ID BELOW AND EXPLN OF PFN DATA SET BELOW
      ID    = PHASE FUNCTION IDENTIFIER
            = 0, USER SUPPLIED
            = 1, MARITIME ARCTIC, VIS=0.1 TO 2.0 KM
            = 2, MARITIME POLAR, VIS=0.2 KM
            = 3, MARITIME POLAR, VIS=0.2, KM
            = 4, CONTINENTAL POLAR, VIS= 0.2 TO 2.5 KM
            = 5, WHITE PHOSPHORUS
            = 6, HEXACHLOROETHANE
            = 7, FOG OIL
            = 8, DUST (MODERATE AEROSOL LOADING)
            = 9, DUST (HEAVY AEROSOL LOADING)
            =10, MARITIME MODEL B, VIS=5KM, RH=95%
            =11, MARITIME MODEL B, VIS=10KM, RH=90%
            =12, MARITIME MODEL B, VIS=50KM, RH=50%
CARD 2  **** IF ISORC LT 2 OR ITARG LT 1 THIS CARD IS NOT NEEDED
EMIS  EM(1), TM(1), EM(2), TM(2)
      EM(1)  EMISSIVITY OF GROUND
      TM(1)  TEMPERATURE OF GROUND (KELVIN)
      EM(2)  EMISSIVITY OF TARGET
      TM(2)  TEMPERATURE OF TARGET (KELVIN)
CARD 3
ATM   ZENTH, CLDHGT, PHASE, BETAEX
      ZENTH  INCIDENT ANGLE OF SUNLIGHT OR MOONLIGHT (DEGREES)
      CLDHGT HEIGHT OF BOTTOM OF CLOUD LAYER (KM)
              ONLY NEEDED WHEN IHAZE NE 0 (DEFAULT IS 0.)

```

SPOT0010
 SPOT0020
 SPOT0030
 SPOT0040
 SPOT0050
 SPOT0060
 SPOT0070
 SPOT0080
 SPOT0090
 SPOT0100
 SPOT0110
 SPOT0120
 SPOT0130
 SPOT0140
 SPOT0150
 SPOT0160
 SPOT0170
 SPOT0180
 SPOT0190
 SPOT0200
 SPOT0210
 SPOT0220
 SPOT0230
 SPOT0240
 SPOT0250
 SPOT0260
 SPOT0270
 SPOT0280
 SPOT0290
 SPOT0300
 SPOT0310
 SPOT0320
 SPOT0330
 SPOT0340
 SPOT0350
 SPOT0360
 SPOT0370
 SPOT0380
 SPOT0390
 SPOT0400
 SPOT0410
 SPOT0420
 SPOT0430
 SPOT0440
 SPOT0450
 SPOT0460
 SPOT0470
 SPOT0480
 SPOT0490
 SPOT0500
 SPOT0510
 SPOT0520
 SPOT0530
 SPOT0540
 SPOT0550
 SPOT0560
 SPOT0570
 SPOT0580
 SPOT0590
 SPOT0600
 SPOT0610
 SPOT0620
 SPOT0630
 SPOT0640
 SPOT0650
 SPOT0660
 SPOT0670
 SPOT0680
 SPOT0690
 SPOT0700

```

C      PHASE      PHASE ANGLE FOR MOONLIGHT (DEGREES)      SPOT0710
C      BETAEX     USER SUPPLIED EXTINCTION COEFFICIENT      SPOT0720
C                                     VALID ONLY WHEN IHAZE=7      SPOT0730
C
C      CARD 4
C      TARG      RTARG,COSX,COSY,COSZ      SPOT0740
C      RTARG     SLANT RANGE FROM RECEIVER TO TARGET (KM)      SPOT0750
C      COSX      X-DIRECTIONAL ANGLE OF TARGET NORMAL (DEGREES)      SPOT0760
C      COSY      Y-DIRECTIONAL ANGLE OF TARGET NORMAL (DEGREES)      SPOT0770
C      COSZ      Z-DIRECTIONAL ANGLE OF TARGET NORMAL (DEGREES)      SPOT0780
C
C      CARD 5      **** IF ITARG LT 1 THIS CARD IS NOT NEEDED      SPOT0790
C      REFL      A0(1),A1(1),IALB(1),A0(2),A1(2),IALB(2)      SPOT0800
C      A0(1)     ALBEDO COEFFICIENT FOR GROUND      SPOT0810
C      A1(1)     ALBEDO COEFFICIENT FOR GROUND      SPOT0820
C      IALB(1)   TYPE OF REFLECTION DISTRIBUTION FOR GROUND      SPOT0830
C      A0(2)     ALBEDO COEFFICIENT FOR TARGET      SPOT0840
C      A1(2)     ALBEDO COEFFICIENT FOR TARGET      SPOT0850
C      IALB(2)   TYPE OF REFLECTION DISTRIBUTION FOR TARGET      SPOT0860
C      IALB      0 LAMBERTIAN REFLECTION SURFACE      SPOT0870
C      1 ISOTROPIC      SPOT0880
C
C      CARD 6
C      SENS      ALT, THETA, PHI, SANG2      SPOT0890
C      ALT       ALTITUDE OF RECEIVER (KM)      SPOT0900
C      THETA     POLAR DIRECTION OF LOOK ANGLE (DEGREES)      SPOT0910
C      PHI       AZMITH DIRECTION OF LOOK ANGLE (DEGREES)      SPOT0920
C
C      *** METEOROLOGICAL AZIMUTH CONVENTION ASSUMED: N = 0 DEG,      SPOT0930
C      *** E = 90 DEG, S = 180 DEG, W = 270 DEG *****      SPOT0940
C
C      SANG2     HALF ANGLE DEFINING RECEIVER FIELD-OF-VIEW      SPOT0950
C      (DEGREES)      SPOT0960
C
C      CARD 7
C      GO        SIGNIFIES TO BEGIN EXECUTION, NO MORE INPUT FOR      SPOT0970
C                  THIS CALL. NOTE THAT IF A DATA CARD IS NOT READ      SPOT0980
C                  THEN ANY VALUES ESTABLISHED FROM PREVIOUS CALLS      SPOT0990
C                  TO THE MODULE ARE STILL IN EFFECT.      SPOT1000
C
C      OPTIONAL CARDS FOR RESPONSE FUNTION (SET BY NR=1 IN EOMAIN)      SPOT1010
C      THESE CARDS MUST FOLLOW THE GO CARD AND CAN ONLY BE INSERTED ONCE      SPOT1020
C      CARD 1: NUMBER OF VALUES FOR RESPONSE FUNCTION - FORMAT (I2).      SPOT1030
C      CARDS 2 - NUMBER OF VALUES: FORMAT (2(E10.4,1X))      SPOT1040
C      ONE VALUE OF WAVE (UM) AND RESPONSE FUNCTION PER CARD      SPOT1050
C      N.B. ONLY ONE RESPONSE FUNCTION PER EOSAEL RUN.      SPOT1060
C
C      AUXILLARY READ FROM UNIT IPHFUN (ASL DATA SET PROVIDED WITH EOSAEL)      SPOT1070
C      ANG       ANGLES AT WHICH PHASE MATRIX IS DEFINED,      SPOT1080
C      NANG      NANG VALUES (DEFAULT IS 65), FORMAT(11(F6.2,1X))      SPOT1090
C      ID,WAVE,OMEGA0,BETAEX,BETABS      SPOT1100
C      NUMBER OF ANGLES AT WHICH THE PHASE FUNCTION HAS      SPOT1110
C      VALUES, PFN IDENTIFIER, WAVELENGTH(UM), SINGLE      SPOT1120
C      SCATTERING ALBEDO, EXTINCTION COEFFICIENTS (TOTAL      SPOT1130
C      AND SCATTERING).      SPOT1140
C      FORMAT (2(I2,1X),F5.2,1X,F8.6,1X,2(E12.6,1X)).      SPOT1150
C      PF        PHASE FUNCTION AT SPECIFIED ANGLES,      SPOT1160
C      FORMAT (6(E12.6,1X))      SPOT1170
C
C      LOGICAL L1,L2,L3,L4,L5,L6,L7,ISPOT,N16,LOREAD,HORIZ      SPOT1180
C      DIMENSION DUMMY(16)      SPOT1190
C      EQUIVALENCE (ITARG,IT)      SPOT1200
C      COMMON /ANSW2/TTR(16),TBR(16),CNTRST(16)      SPOT1210
C      COMMON /ALBED/A0(2),A1(2),IALB(2)      SPOT1220
C      COMMON /BKDAT/ALT,THETA,PHI,SANG2,ZENTH,PHASE,ALB      SPOT1230
C      COMMON /CGEOM/COSGM,COSBT,COSIN      SPOT1240
C      COMMON /COM11/ISORC,ITARG,IWN,JHL      SPOT1250
C      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK      SPOT1260
C      COMMON /CTARG/RTARG,COSX,COSY,COSZ      SPOT1270
C      COMMON /EMISS/EM(2),TM(2)      SPOT1280
C      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUS      SPOT1290
C      COMMON /MO1/EH(16,34),P(34),T(34),WH(34),Z(34),WA(34),RE,M,NL,      SPOT1300
C      +      RRS(16,34),SCOE(16,34),      SPOT1310
C      +      TRANS(16,3),RADA(16,2),WAVE(16),SS(16),      SPOT1320
C      SPOT1330
C      SPOT1340
C      SPOT1350
C      SPOT1360
C      SPOT1370
C      SPOT1380
C      SPOT1390
C      SPOT1400

```

```

1 DIR(16), RADG(16), UTEM(16), UTRF(16), BK(16), SPOT1410
2 PATHR(16), UERF(16), PATHR2(16), TOT(12), BKG(16) SPOT1420
COMMON /MO2/ WO(34), RO, TBOUND, JP, IM, ML, IP, JSTOR SPOT1430
COMMON /EM2/W(16), E(16), IL, IKMAX, LENTOR, NLL SPOT1440
COMMON /SPOTLO/ ISPOT, LOREAD, N16 SPOT1450
COMMON /LOWEX/ WPATH(68,16), WLAY(34,16), TBBY(68), TX(16), BETAEX, SPOT1460
1 CLDHGT, NCLD SPOT1470
COMMON /BASPOT/ ANG(65), SUM(65), WVL(16), NWVL, ALBB(16), BS(16), SPOT1480
1 BE(16), SINGWV, PF(65), LMAX SPOT1490
COMMON /LOGIC/L1, L2, L3, L4, L5, L6, L7 SPOT1500
DATA L1, L2, L3, L4, L5, L6, L7/7*.FALSE./ SPOT1510
DATA ITR1, ITR2, ITR3, ITR4/2,3,5,1/ SPOT1520
N16=.TRUE. SPOT1530
INBP=16 SPOT1540
DUM=1. SPOT1550
ICLMAT=0 SPOT1560
LOREAD=.TRUE. SPOT1570
ISPOT=.TRUE. SPOT1580
C INITIALIZE AND READ INPUT PARAMETERS SPOT1590
CALL ZERO SPOT1600
CALL INDAT(IEMISS, IHAZE, IM, LEN, ML, MODEL, SPOT1610
1 TBOUND, 0, CLDHGT, BETAEX) SPOT1620
C CHECK FOR ERROR IN INPUT DATA SPOT1630
IF (IHAZE.EQ.9) IERR=1 SPOT1640
IF (IERR.EQ.1) RETURN SPOT1650
IF (ISORC.NE.2) ISWTCH=1 SPOT1660
C FIRST CALL IS TO READ LOWTRAN DATA FILE ONLY SPOT1670
CALL LT4M(ALT, DUM, ZENTH, 3, 0, TRANS(1,1), DUMMY, DUMMY, SPOT1680
1 IEMISS, LEN, MODEL, VIS, V1, V2, TGRD, SPOT1690
2 ICLMAT, IERR, NR, IHAZE, MULDV) SPOT1700
V1=WAVN1 SPOT1710
V2=WAVN2 SPOT1720
CALL CKER(V1, V2, DV, IV1, IV2, IDV, IERR, MULDV, ISPOT, DUM) SPOT1730
WAVE(1)=10000./V1 SPOT1740
DO 300 IW=2, INBR SPOT1750
V2=V1+20.*FLOAT(MULDV)*FLOAT(IW-1) SPOT1760
IF (V2.GE.WAVN2) GO TO 400 SPOT1770
WAVE(IW)=10000./V2 SPOT1780
300 CONTINUE SPOT1790
L1=.TRUE. SPOT1800
C MAXIMUM NO. OF WAVELENGTHS SPOT1810
400 IWN=IW-1 SPOT1820
IF (L1) IWN=INBR SPOT1830
NWVL=IWN SPOT1840
C ARRAY WVL IS USED ONLY IN SUBROUTINE PFUNC. THE WAVELENGTHS SPOT1850
IN THIS ARRAY INCREASE WITH INCREASING ARRAY INDEX. THE SPOT1860
VALID RANGES FOR VALUES IN THE WVL ARRAY ARE : 0.2-2.0, 3.0-5.0, SPOT1870
AND 8.0-12.0 MICROMETERS. SPOT1880
C SPOT1890
DO 355 JX=1, NWVL SPOT1900
INDM=NWVL-JX+1 SPOT1910
355 WVL(JX)=WAVE(INDM) SPOT1920
DO 500 I=1, NL SPOT1930
IF (ALT.LE.Z(I)) GO TO 600 SPOT1940
500 CONTINUE SPOT1950
WRITE (IOOUT,3700) ALT, I, Z(I) SPOT1960
IERR=1 SPOT1970
RETURN SPOT1980
600 CONTINUE SPOT1990
IF (I.EQ.1) WRITE (IOOUT,3800) ALT SPOT2000
IF (I.EQ.1) IERR=1 SPOT2010
IF (IERR.EQ.1) RETURN SPOT2020
JHL=I-1 SPOT2030
NLL=NL-1 SPOT2040
SANG=TWOP1*(1.0-COS(SANG2*PIRAD)) SPOT2050
CZNTN=COS(ZENTH*PIRAD) SPOT2060
SZNTN=SIN(ZENTH*PIRAD) SPOT2070
CTHTA=COS(THETA*PIRAD) SPOT2080
STHTA=SIN(THETA*PIRAD) SPOT2090
SPOT2100

```

```

C      CPHI=COS(PHI*PIRAD)
C      SPHI=SIN(PHI*PIRAD)
C      IF (ISORC.EQ.2) GO TO 900
C      CALCULATE SOURCE TERM ... FOR SUNLIGHT (ISORC=0,3)
C                                     OR MOONLIGHT (ISORC=1,4)
      DO 700 IW=1,IWN
      SS(IW)=0.0
      IF (ISORC.EQ.0.OR.ISORC.EQ.3) SS(IW)=SOLARS(WAVE(IW))
      IF (ISORC.EQ.1.OR.ISORC.EQ.4) SS(IW)=SMOON(WAVE(IW),PHASE)
700  CONTINUE
C      CALCULATE DIRECT INTENSITY ... FOR SUNLIGHT (ISORC=0,3)
C                                     OR MOONLIGHT (ISORC=1,4)
      COSIN=STHTA*CPHI*SZNTH+CTHTA*CZNTH
      ANGIN=ACOS(COSIN)/PIRAD
      READ PHASE FUNCTION FILE
      CALL INDAT(IEMISS,IHAZE,IM,LEN,ML,MODEL,
1      TBOUND,1,CLDHGT,BETAEX)
      IF (ANGIN.GT.SANG2) L2=.TRUE.
      IF (ZENTH.GT.80.0) L3=.TRUE.
      CALL LT4M(ALT,DUM,ZENTH,3,0,TRANS(1,1),DUMMY,DUMMY,
1      IEMISS,LEN,MODEL,VIS, V1,V2,TGRD,
2      ICLMAT,IERR,NR,IHAZE,MULDV)
      IF (L2.OR.L3) GO TO 900
      DO 800 IW=1,IWN
      DIR(IW)=SS(IW)*TRANS(IW,1)
800  CONTINUE
900  IF (ITARG.EQ.0.AND.THETA.GT.90.0) L7=.TRUE.
      IF (L7) GO TO 3500
      IF (ITARG.EQ.0) GO TO 1200
      IF (ITARG.EQ.1) GO TO 1000
C      TARGET...
      ZTARG=RTARG*CTHTA+ALT
      COSTX=STHTA*CPHI
      COSTY=STHTA*SPHI
      COSTZ=CTHTA
      COSBT=COSX*SZNTH+COSZ*CZNTH
      COSGM=-(COSTX*COSX+COSTY*COSY+COSTZ*COSZ)
      IF (THETA.LE.90.0) GO TO 1100
      COSBTG=CZNTH
      COSGMG=COS((180.0-THETA)*PIRAD)
      GO TO 1100
C      GROUND...
1000 ZTARG=0.0
      COSBT=CZNTH
      COSGM=COS((180.0-THETA)*PIRAD)
      COSBTG=COSBT
      COSGMG=COSGM
C      CALCULATE ATMOSPHERIC TRANSMISSION/RADIANCE FOR VARIOUS PATHS
1100 IF (COSGM.LE.0.0) L4=.TRUE.
      IF (COSBT.LE.0.0) L5=.TRUE.
      IF (THETA.LE.90.0.AND.ITARG.EQ.1) L6=.TRUE.
      IF (L4.OR.L6) GO TO 3500
1200 CONTINUE
      IF (THETA.EQ.90.0)
1      CALL LT4M(ALT,DUM,1000.0,1,2,TRANS(1,4),RADA(1,1),
2      DUMMY, IEMISS,LEN,MODEL,VIS, V1,V2,
3      TGRD,ICLMAT,IERR,NR,IHAZE,MULDV)
      IF (THETA.EQ.90.0.AND.ITARG.EQ.2)
1      CALL LT4M(ALT,DUM,RTARG,1,2,TRANS(1,2),RADA(1,2),
2      DUMMY, IEMISS,LEN,MODEL,VIS, V1,V2,
3      TGRD,ICLMAT,IERR,NR,IHAZE,MULDV)
      IF (THETA.EQ.90.0) GO TO 1300
      IF (THETA.LT.90.0)
1      CALL LT4M(ALT,DUM,THETA,3,2,TRANS(1,4),RADA(1,2),DUMMY,
2      IEMISS,LEN,MODEL,VIS, V1,V2,TGRD,
3      ICLMAT,IERR,NR,IHAZE,MULDV)
      IF (ITARG.EQ.0) GO TO 2000
      IF (THETA.LT.90.0)
1      CALL LT4M(ALT,ZTARG,THETA,2,2,TRANS(1,2),RADA(1,2)
2      ,DUMMY, IEMISS,LEN,MODEL,VIS, V1,V2,

```

```

SPOT2110
SPOT2120
SPOT2130
SPOT2140
SPOT2150
SPOT2160
SPOT2170
SPOT2180
SPOT2190
SPOT2200
SPOT2210
SPOT2220
SPOT2230
SPOT2240
SPOT2250
SPOT2260
SPOT2270
SPOT2280
SPOT2290
SPOT2300
SPOT2310
SPOT2320
SPOT2330
SPOT2340
SPOT2350
SPOT2360
SPOT2370
SPOT2380
SPOT2390
SPOT2400
SPOT2410
SPOT2420
SPOT2430
SPOT2440
SPOT2450
SPOT2460
SPOT2470
SPOT2480
SPOT2490
SPOT2500
SPOT2510
SPOT2520
SPOT2530
SPOT2540
SPOT2550
SPOT2560
SPOT2570
SPOT2580
SPOT2590
SPOT2600
SPOT2610
SPOT2620
SPOT2630
SPOT2640
SPOT2650
SPOT2660
SPOT2670
SPOT2680
SPOT2690
SPOT2700
SPOT2710
SPOT2720
SPOT2730
SPOT2740
SPOT2750
SPOT2760
SPOT2770
SPOT2780
SPOT2790
SPOT2800

```

```

3   TGRD,ICLMAT,IERR,NR,IHAZE,MULDV)
   IF (THETA.LT.90.0) GO TO 1300
   CALL LT4M(ALT,ZTARG,THETA,2,2,TRANS(1,2),RADA(1,2),
1   RADG(1),IEMISS,LEN,MODEL,VIS,V1,V2,
2   TGRD,ICLMAT,IERR,NR,IHAZE,MULDV)
   IF (ITARG.NE.2) ITR4=2
   IF (ITARG.NE.2) GO TO 1300
   IF (ZTARG.LE.0.0) ITR4=2
   IF (ZTARG.LE.0.0) GO TO 1300
   ITR1=4
   CALL LT4M(ALT,0.0,THETA,2,2,TRANS(1,4),RADA(1,1),
1   RADG(1),IEMISS,LEN,MODEL,VIS,V1,V2,
2   TGRD,ICLMAT,IERR,NR,IHAZE,MULDV)
1300 IF (ISORC.LT.2) GO TO 1600
C   CALCULATE UNCOLLIDED EMISSION ... FROM GROUND (ITARG=1)
C   OR TARGET (ITARG=2)
   DO 1500 IW=1,IWN
   WAVEM=WAVE(IW)/1.0E+4
   IF (ITARG.EQ.2) GO TO 1400
   BK(IW)=BLACK(WAVEM,TGRD)*EM(1)
   RADG(IW)=RADG(IW)*EM(1)*COSGMG
   BK(IW)=BK(IW)
   UTEM(IW)=RADG(IW)
   GO TO 1500
1400 BK(IW)=BLACK(WAVEM,TM(2))*EM(2)
   UTEM(IW)=BK(IW)*COSGM*TRANS(IW,2)
   IF (THETA.LE.90.0) GO TO 1500
   BK(IW)=BLACK(WAVEM,TGRD)*EM(1)
   RADG(IW)=RADG(IW)*EM(1)*COSGMG
1500 CONTINUE
1600 IF (ISORC.EQ.2) GO TO 2900
C   CALCULATE UNCOLLIDED REFLECTANCE ... FROM GROUND (ITARG=1)
C   OR TARGET (ITARG=2)
   IEMISS=0
   IF (L5) GO TO 2000
   HORIZ=ABS(ZTARG-ALT).LT.0.001
   IF (HORIZ) ITR2=1
   IF (HORIZ) GO TO 1700
   CALL LT4M(ZTARG,DUM,ZENTH,3,2,TRANS(1,3),DUMMY,DUMMY,
1   IEMISS,LEN,MODEL,VIS,V1,V2,TGRD,
2   ICLMAT,IERR,NR,IHAZE,MULDV)
1700 IF (ZTARG.LE.0.0) ITR3=ITR2
   IF (ZTARG.LE.0.0) GO TO 1800
   IF (THETA.LE.90.0) GO TO 1800
   CALL LT4M(0.0,DUM,ZENTH,3,2,TRANS(1,5),DUMMY,DUMMY,
1   IEMISS,LEN,MODEL,VIS,V1,V2,TGRD,
2   ICLMAT,IERR,NR,IHAZE,MULDV)
1800 DO 1900 IW=1,IWN
   ALB=ALBEDO(IT)
   UTRF(IW)=SS(IW)*COSBT*ALB*TRANS(IW,2)*TRANS(IW,ITR2)
   IF (THETA.LE.90.0) GO TO 1900
   UERF(IW)=SS(IW)*COSBTG*ALBEDO(1)*TRANS(IW,ITR1)*
1   TRANS(IW,ITR3)
1900 CONTINUE
C   CALCULATE SINGLE-SCATTERED PATH RADIANCE ...
C   FROM SUNLIGHT (ISORC=0,3)
C   OR MOONLIGHT (ISORC=1,4)
2000 CALL COEFS(P,T,VIS,IHAZE,ZTARG,NCLD,IERR,BETAEX)
   IF (IERR.EQ.1) RETURN
   IF (ITARG.EQ.0) GO TO 2700
   CALL PATHRD(CTHTA,ALT,RTARG,1,IHAZE,NR,
1   IEMISS,LEN,MODEL,VIS,V1,V2,TGRD,DUMMY,ICLMAT,MULDV)
   DO 2100 IW=1,IWN
2100 PATHR2(IW)=SS(IW)*PATHR(IW)
   IF (ABS(CTHTA).LE.1.0E-3) GO TO 2600
   IF (CTHTA.LT.0.0) GO TO 2300
2200 Z=Z(NLL)
   IO=2
   RT=(Z(NLL)-ZTARG)/CTHTA
   GO TO 2800
SPOT2810
SPOT2820
SPOT2830
SPOT2840
SPOT2850
SPOT2860
SPOT2870
SPOT2880
SPOT2890
SPOT2900
SPOT2910
SPOT2920
SPOT2930
SPOT2940
SPOT2950
SPOT2960
SPOT2970
SPOT2980
SPOT2990
SPOT3000
SPOT3010
SPOT3020
SPOT3030
SPOT3040
SPOT3050
SPOT3060
SPOT3070
SPOT3080
SPOT3090
SPOT3100
SPOT3110
SPOT3120
SPOT3130
SPOT3140
SPOT3150
SPOT3160
SPOT3170
SPOT3180
SPOT3190
SPOT3200
SPOT3210
SPOT3220
SPOT3230
SPOT3240
SPOT3250
SPOT3260
SPOT3270
SPOT3280
SPOT3290
SPOT3300
SPOT3310
SPOT3320
SPOT3330
SPOT3340
SPOT3350
SPOT3360
SPOT3370
SPOT3380
SPOT3390
SPOT3400
SPOT3410
SPOT3420
SPOT3430
SPOT3440
SPOT3450
SPOT3460
SPOT3470
SPOT3480
SPOT3490
SPOT3500

```

2300	IF (ZTARG.GT.0.0) GO TO 2500	SPOT3510
	DO 2400 IW=1,IWN	SPOT3520
2400	PATHR(IW)=0.0	SPOT3530
	GO TO 2900	SPOT3540
2500	Z2=0.0	SPOT3550
	IO=1	SPOT3560
	RT=-ZTARG/CTHTA	SPOT3570
	GO TO 2800	SPOT3580
2600	Z2=ZTARG	SPOT3590
	RT=1000.0	SPOT3600
	IO=3	SPOT3610
	GO TO 2800	SPOT3620
2700	IEMISS=0	SPOT3630
	ZTARG=ALT	SPOT3640
	IF (ABS(CTHTA).LE.1.0E-03) GO TO 2600	SPOT3650
	GO TO 2200	SPOT3660
2800	CALL PATHRD(CTHTA,ZTARG,RT,IO,IHAZE,NR,	SPOT3670
	IEMISS,LEN,MODEL,VIS,V1,V2,TGRD,DUMMY,ICLMAT,MULDV)	SPOT3680
C	CALCULATE BACKGROUND AND TOTAL INTENSITY, PLUS	SPOT3690
C	CONTRAST RATIO	SPOT3700
2900	DO 3100 IW=1,IWN	SPOT3710
	PATHR(IW)=PATHR2(IW)+SS(IW)*PATHR(IW)	SPOT3720
	TTR(IW)=PATHR2(IW)+UTRF(IW)+RADA(IW,2)+UTEM(IW)	SPOT3730
	TBR(IW)=PATHR(IW)+UERF(IW)+RADA(IW,1TR4)+RADG(IW)	SPOT3740
	DIF=TTR(IW)-TBR(IW)	SPOT3750
	IF (TBR(IW).GT.0.0) GO TO 3000	SPOT3760
	IF (TTR(IW).EQ.TBR(IW)) CNTRST(IW)=0.0	SPOT3770
	IF (TTR(IW).NE.TBR(IW)) CNTRST(IW)=1.0E30	SPOT3780
	GO TO 3100	SPOT3790
3000	CNTRST(IW)=DIF/TBR(IW)	SPOT3800
3100	CONTINUE	SPOT3810
C	CALCULATE TOTAL RADIANCES INTEGRATED OVER DETECTOR RESPONSE	SPOT3820
	DV2=DV*0.5	SPOT3830
	SUMRPF=0.	SPOT3840
	DO 3200 IW=1,IWN	SPOT3850
	NW=10000./WAVE(IW)	SPOT3860
	W2=10000./((FLOAT(NW)-DV2)	SPOT3870
	W1=10000./((FLOAT(NW)+DV2)	SPOT3880
	IF (IW.EQ.1) W2=WAVE(1)	SPOT3890
	IF (IW.EQ.IWN) W1=WAVE(IWN)	SPOT3900
	RESPFN=RESFN(NR,WAVE(IW))	SPOT3910
	SUMRPF=SUMRPF+RESPFN	SPOT3920
	DW=(W2-W1)*RESPFN	SPOT3930
	TOT(1)=TOT(1)+DW*UTEM(IW)	SPOT3940
	TOT(2)=TOT(2)+DW*UTRF(IW)	SPOT3950
	TOT(3)=TOT(3)+DW*RADA(IW,2)	SPOT3960
	TOT(4)=TOT(4)+DW*PATHR2(IW)	SPOT3970
	TOT(5)=TOT(5)+DW*TTR(IW)	SPOT3980
	TOT(6)=TOT(6)+DW*RADG(IW)	SPOT3990
	TOT(7)=TOT(7)+DW*UERF(IW)	SPOT4000
	TOT(8)=TOT(8)+DW*RADA(IW,1TR4)	SPOT4010
	TOT(9)=TOT(9)+DW*PATHR(IW)	SPOT4020
	TOT(10)=TOT(10)+DW*TBR(IW)	SPOT4030
3200	TOT(11)=TOT(11)+DW*DIR(IW)	SPOT4040
	IF (NR.NE.1) SUMRPF=1.	SPOT4050
	DO 3250 I=1,11	SPOT4060
3250	TOT(I)=TOT(I)/SUMRPF	SPOT4070
	IF (TOT(10).GT.0.0) GO TO 3300	SPOT4080
	IF (TOT(5).EQ.TOT(10)) TOT(12)=0.0	SPOT4090
	IF (TOT(5).NE.TOT(10)) TOT(12)=1.0E30	SPOT4100
	GO TO 3400	SPOT4110
3300	TOT(12)=(TOT(5)-TOT(10))/TOT(10)	SPOT4120
3400	CONTINUE	SPOT4130
3500	CALL OUTPUT(MODEL,IHAZE,CLDHGT)	SPOT4140
	RETURN	SPOT4150
C		SPOT4160
3700	FORMAT (1H,11H ALTITUDE (<F10.3,17H) GREATER THAN Z<,	SPOT4170
1	12,2H)=,F10.3,27H CONTROL RETURNED TO MAIN	SPOT4180
2	10HFROM SPOT.)	SPOT4190
3800	FORMAT (1H,11H ALTITUDE (<F10.3,16H) LESS THAN ZERO,	SPOT4200

1 37H CONTROL RETURNED TO MAIN FROM SPOT. >
END

SPOT4210
SPOT4220

C	FUNCTION ALBEDO(I)	ALB00010
C	COMMON /ALBED/A0(2),A1(2),IALB(2)	ALB00020
C	COMMON /CGEOM/COSGM,COSBT,COSIN	ALB00030
	COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK	ALB00040
	CALCULATE ALBEDO FOR GROUND (ITARG=1) OR TARGET (ITARG=2)	ALB00050
	IALB(I) = 0 LAMBERTIAN REFLECTION SURFACE	ALB00060
	1 ISOTROPIC	ALB00070
	A=A0(I)+A1(I)*COSBT	ALB00080
	IF (IALB(I).EQ.0) ALBEDO=A*COSGM/PI	ALB00090
	IF (IALB(I).EQ.1) ALBEDO=A/TWOPI	ALB00100
	RETURN	ALB00110
	END	ALB00120

CCCCC

FUNCTION BLACK(W,T)
 BLACK(W,T) = PLANCK FUNCTION (UNITS: WATT PER SQUARE METER PER
 MICROMETER PER STERADIAN), GIVEN WAVELENGTH W IN CM AND TEMP-
 ERATURE T IN K

EXP OVERFLOW PROTECTION

ARG=1.43879/(W*T)
 IF(ARG.LT.88.) GO TO 1
 BLACK=0.0
 RETURN
 1 BLACK=1.19106E-12/(W**5*(EXP(ARG)-1.0))
 RETURN
 END

BLA00010
 BLA00020
 BLA00030
 BLA00040
 BLA00050
 BLA00060
 BLA00070
 BLA00080
 BLA00090
 BLA00100
 BLA00110
 BLA00120
 BLA00130
 BLA00140

```

SUBROUTINE COEFS(P,T,VIS,IHAZE,ZTARG,NCLD,IERR,BETAEX)
COMMON /BKDAT/ALT,THETA,PHI,SANG2,ZENTH,PHASE,ALB
COMMON /CTARG/RTARG,COSX,COSY,COSZ
COMMON /MO1/DUMMIE(715),MHOLD,NL,
+      RRS(16,34),SCOE(16,34),
+      TRANS(16,5),RADA(16,2),WAVE(16),SS(16),
1      DIR(16),RADG(16),UTEM(16),UTRF(16),BK(16),
2      PATHR(16),UERF(16),PATHR2(16),TOT(12),BKG(16)
COMMON /BASPT/ ANG(65),SUM(65),WVL(16),NWVL,ALBB(16),BS(16),
1 BE(16),SINGWV,PF(65)
COMMON /COM1/ISORC,ITARG,IWN,JHL
COMMON /CONST/ PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK
DIMENSION P(34),T(34)
C      CALCULATE THE WAVELENGTH-DEPENDENT CONSTANT PRESSURE COEFFICIENTS
C      FOR MOLECULAR SCATTERING.
C      LOOP OVER LAYERS
DO 600 I=1,NL
PS=P(I)/1013.0
TS=CDEGK/T(I)
RSCAT=PS*TS
C      LOOP OVER WAVELENGTHS
DO 600 IW=1,IWN
RAYS=0.0
NW=10000./WAVE(IW)
C      RAYLEIGH SCATTERING = 0.FOR WAVELENGTH GT 3.33 UM
IF (NW.LT.3000) GO TO 200
WN=FLOAT(NW)
RAYS=RSCAT*(WN**4)/(9.67578E+18-1.11836E+09*WN**2)
200 CONTINUE
AEXT=1.
AABS=1.
IF (IHAZE.EQ.0.OR.I.NE.(NCLD-1)) GO TO 1
EXT55=3.912/VIS
C      UPPER LIMIT OF 500 METERS VERTICAL DISTANCE FOR XSCALE
ZTALT=ZTARG/ALT
IF (ABS(ZTALT-1.)>.01) RNG=RTARG
IF ((ZTARG.GT.ALT).AND.(RTARG.GT..5/COS(THETA*PIRAD)))
1 RNG=.5/COS(THETA*PIRAD)
IF (ZTARG.LT.RTARG.AND.(RTARG.GT..5/COS((180.-THETA)*PIRAD)))
1 RNG=.5/COS((180.-THETA)*PIRAD)
IF (ITARG.EQ.0.AND.(RTARG.GT..5/COS(THETA*PIRAD)))
1 RNG=.5/COS(THETA*PIRAD)
ISLANT=1
IF (ABS(ZTALT-1.)>.01) ISLANT=0
C      CALL XSCALE FOR TOTAL PATH LENGTH TRANSMISSION FOR AEROSOL
CALL XSCALE(WAVE(IW),88.,EXT55,XSTRN,IERR,ISLANT,IHAZE,RNG,THETA)
IF (IERR.EQ.1) RETURN
AEXT=-ALOG(XSTRN)/RNG
C      USER SUPPLIED COEFF(IHAZE=7), OR READ FROM PFN DATA FILE(IHAZE=8)
IF (IHAZE.EQ.7) AEXT=BETAEX
IF (IHAZE.EQ.8) AEXT=BE(IW)
IF (WAVE(IW).LT.2.) AABS=1.
IF (WAVE(IW).GE.3..AND.WAVE(IW).LE.5.) AABS=AEXT*.2
IF (WAVE(IW).GE.8..AND.WAVE(IW).LE.12.) AABS=AEXT*.45
1 CONTINUE
SCOE(IW,I)=AEXT-AABS+RAYS
C      CHECK FOR NO AEROSOL PRESENT
IF (SCOE(IW,I).LT.1.E-20) RRS(IW,I)=1.0
C      AEROSOL AND RAYLEIGH PRESENT
IF (SCOE(IW,I).GE.1.E-20) RRS(IW,I)=RAYS/SCOE(IW,I)
C      CHECK FOR NO RAYLEIGH SCATTERING
IF (RAYS.LT.1.E-20) RRS(IW,I)=0.0
600 CONTINUE
RETURN
END

```

```

SUBROUTINE DIAG
  THIS SUBROUTINE PRODUCES DIAGNOSTIC COMMENTARY FOR THE
  SPOT PROGRAM.

  CALLING SEQUENCE: CALL DIAG

  EXTERNAL VARIABLES REQUIRED:
    THETA <COMMON BLOCK BKDAT>
    HLIGHT, HTYPE <COMMON BLOCK HOLRTH>
    L1,L2,L3,L4,L5,L6 <COMMON BLOCK LOGIC>

  LOGICAL L1,L2,L3,L4,L5,L6,L7
  COMMON /BKDAT/ALT,THETA,PHI,SANG2,ZENTH,PHASE,ALB
  COMMON /HOLRTH/ HITARG(8,3),HISORC(6,5),
1  HMODEL(5,6),HLIGHT(3),HMNLT(3),HSNLT(3),
2  HTRGT(2),HTYPE(2),HGRND(2)
  COMMON /LOGIC/L1,L2,L3,L4,L5,L6,L7
  COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU
  DATA IZ/1/

  WRITE HEADING,
  WRITE (IOOUT,900)
  IF ERRORS, GO TO 5; OTHERWISE PRINT CLEAN RUN MESSAGE.
  IF (L1.OR.L2.OR.L3.OR.L4.OR.L5.OR.L6.OR.L7) GO TO 100
  WRITE (IOOUT,1000)
  GO TO 700

  HERE IF THERE WERE ERRORS
100 WRITE (IOOUT,1100)
  IF (.NOT.L1) GO TO 200
  WRITE (IOOUT,1200) I
  I=I+1
200 IF (.NOT.L2) GO TO 300
  WRITE (IOOUT,1300) I,HLIGHT
  I=I+1
300 IF (.NOT.L3) GO TO 400
  WRITE (IOOUT,1400) I,HLIGHT
  I=I+1
400 IF (.NOT.L4) GO TO 500
  WRITE (IOOUT,1500) I,HTYPE
  I=I+1
500 IF (.NOT.L5) GO TO 600
  WRITE (IOOUT,1600) I,HLIGHT
  I=I+1
600 IF (.NOT.L6) GO TO 700
  WRITE (IOOUT,1700) I,THETA
  I=I+1
700 IF (.NOT.L7) GO TO 800
  WRITE (IOOUT,1800) I,THETA
  WRITE FOOTING,
800 WRITE (IOOUT,1900)

  900 FORMAT (1H0,21X,90(1H*),3(/,21X,1H*,88X,1H*))
1000 FORMAT (21X,1H*,28X,29HNO SPOT DIAGNOSTICS FOR THIS
1  3HRUN,28X,1H*,/,21X,1H*,28X,8H--- ----,11(1H-)
2  13H --- ----,26X,1H*)
1100 FORMAT (21X,1H*,28X,25HSPOT DIAGNOSTIC MESSAGES
1  7HFOLLOW:,28X,1H*,/,21X,1H*,28X,5H----,10(1H-
2  ),1X,8(1H-),1X,6(1H-),29X,1H*,2(/,21X,1H*,88X,
3  1H*)
1200 FORMAT (21X,1H*,9X,I1,30H. NUMBER OF WAVELENGTHS <IWN>
1  9H EXCEEDS,31HALLOWABLE DIMENSIONS; IWN RESET
2  8X,1H*,/,21X,1H*,13X,6HTO 16.,69X,1H*,/,21X,
3  1H*,88X,1H*)
1300 FORMAT (21X,1H*,9X,I1,12H. NO DIRECT,3A4,9HINCIDENT
1  7HWITHIN,25HRECEIVER'S FIELD OF VIEW.,13X,1H*
2  /,21X,1H*,88X,1H*)
1400 FORMAT (21X,1H*,9X,I1,25H. ANGLE OF INCIDENCE FOR,
1  3A4,8HGREATER,21HTHAN 80.0 DEGREES; NO,12X,
2  1H*,/,21X,1H*,13X,26HCALCULATIONS WILL BE MADE
3  14HFOR ITARG = 0.,35X,1H*,/,21X,1H*,88X,1H*)
1500 FORMAT (21X,1H*,9X,I1,1H.,1X,2A4,14HDOES NOT FACE

```

DIAG0010
DIAG0020
DIAG0030
DIAG0040
DIAG0050
DIAG0060
DIAG0070
DIAG0080
DIAG0090
DIAG0100
DIAG0110
DIAG0120
DIAG0130
DIAG0140
DIAG0150
DIAG0160
DIAG0170
U DIAG0180
DIAG0190
DIAG0200
DIAG0210
DIAG0220
DIAG0230
DIAG0240
DIAG0250
DIAG0260
DIAG0270
DIAG0280
DIAG0290
DIAG0300
DIAG0310
DIAG0320
DIAG0330
DIAG0340
DIAG0350
DIAG0360
DIAG0370
DIAG0380
DIAG0390
DIAG0400
DIAG0410
DIAG0420
DIAG0430
DIAG0440
DIAG0450
DIAG0460
DIAG0470
DIAG0480
DIAG0490
DIAG0500
DIAG0510
DIAG0520
DIAG0530
DIAG0540
DIAG0550
DIAG0560
DIAG0570
DIAG0580
DIAG0590
DIAG0600
DIAG0610
DIAG0620
DIAG0630
DIAG0640
DIAG0650
DIAG0660
DIAG0670
DIAG0680
DIAG0690
DIAG0700

```

1 9HRECEIVER,,45X,1H*,/,21X,1H*,88X,1H*)
1600 FORMAT (21X,1H*,9X,11,1H,,3A4,22HILLUMINATES BACK SIDE
1 3HOF,7HTARGET,,33X,1H*,/,21X,1H*,88X,1H*)
1700 FORMAT (21X,1H*,9X,11,30H, THETA LESS THAN 90 DEGREES
1 10HAND ITARG 13H = 1; THETA =,F6.4,19X,1H*,/
2 21X,1H*,88X,1H*)
1800 FORMAT (21X,1H*,9X,11,25H, THETA GREATER THAN 90
1 13HDEGREES AND 19HITARG = 0; THETA =,F7.4,
2 13X,1H*,/,21X,1H*,88X,1H*)
1900 FORMAT (2(21X,1H*,88X,1H*,/),21X,90(1H*))
RETURN
END

```

```

DIAG0710
DIAG0720
DIAG0730
DIAG0740
DIAG0750
DIAG0760
DIAG0770
DIAG0780
DIAG0790
DIAG0800
DIAG0810
DIAG0820

```


	IF (DAT(1).EQ.TYPE(5)) GO TO 5	IND00710
	IF (DAT(1).EQ.TYPE(6)) GO TO 6	IND00720
	IF (DAT(1).EQ.TYPE(7)) GO TO 7	IND00730
C	ERROR RETURN	IND00740
	WRITE (IOOUT,101)	IND00750
101	FORMAT(1H,48HINCORRECT INPUT CARD FOR SPOT, CONTROL RETURNED ,	IND00760
	1,18HTO MAIN FROM INDAT)	IND00770
	IHAZE=9	IND00780
	RETURN	IND00790
C	OPERATING ENVIRONMENT	IND00800
1	ISORC=IFIX(DAT(2))	IND00810
	ITARG=IFIX(DAT(3))	IND00820
	IHAZE=IFIX(DAT(4))	IND00830
	MODEL=IFIX(DAT(5))	IND00840
	NLAM=IFIX(DAT(6))	IND00850
	ID=IFIX(DAT(7))	IND00860
	GO TO 8	IND00870
C	EMISSIONITY AND TEMPERATURE OF GROUND AND TARGET, RESPECTIVELY	IND00880
2	EM(1)=DAT(2)	IND00890
	TM(1)=DAT(3)	IND00900
	EM(2)=DAT(4)	IND00910
	TM(2)=DAT(5)	IND00920
	TBOUND=TM(1)	IND00930
	GO TO 8	IND00940
C	INCIDENT ZENITH ANGLE OF RADIATION, CLOUD BOTTOM HEIGHT,	IND00950
3	PHASE ANGLE OF MOON, OPTIONAL EXTN COEF (VALID WHEN IHAZE=8)	IND00960
	ZENTH=DAT(2)	IND00970
	CLDHGT=DAT(3)	IND00980
	PHASE=DAT(4)	IND00990
	BETAEX=DAT(5)	IND01000
	GO TO 8	IND01010
C	TARGET PROPERTIES	IND01020
4	RTARG=DAT(2)	IND01030
	COSX=DAT(3)	IND01040
	COSY=DAT(4)	IND01050
	COSZ=DAT(5)	IND01060
	GO TO 8	IND01070
C	ALBEDO COEFFICIENTS AND TYPE OF REFLECTION SURFACE FOR	IND01080
5	GROUND AND TARGET, RESPECTIVELY	IND01090
	A0(1)=DAT(2)	IND01100
	A1(1)=DAT(3)	IND01110
	IALB(1)=IFIX(DAT(4))	IND01120
	A0(2)=DAT(5)	IND01130
	A1(2)=DAT(6)	IND01140
	IALB(2)=IFIX(DAT(7))	IND01150
	GO TO 8	IND01160
C	SENSOR CHARACTERISTICS	IND01170
6	ALT=DAT(2)	IND01180
	THETA=DAT(3)	IND01190
	PHI=DAT(4)	IND01200
C	EXPECTING INPUT AZIMUTH IN METEOROLOGICAL CONVENTION	IND01210
C	(I.E., N = 0 DEG, E = 90 DEG, S = 180 DEG, W = 270 DEG),	IND01220
C	SO CONVERT TO MATHEMATICAL CONVENTION FOR PURPOSES OF	IND01230
C	SPOT (ASSUMING Y-AXIS IS POSITIVE NORTHWARD, X-AXIS POSI-	IND01240
C	TIVE EASTWARD).	IND01250
	PHI=90.-PHI	IND01260
	SANG2=DAT(5)	IND01270
8	CONTINUE	IND01280
10	CONTINUE	IND01290
7	CONTINUE	IND01300
	IEMISS=0	IND01310
	IM=0	IND01320
	LEN=0	IND01330
	ML=0	IND01340
	IF (ISORC.GT.1) IEMISS=1	IND01350
	COSX=COS(COSX*PIRAD)	IND01360
	COSY=COS(COSY*PIRAD)	IND01370
	COSZ=COS(COSZ*PIRAD)	IND01380
		IND01390
		IND01400

C	GEOMETRICAL OPTION	IND01410
	IF (IGEOSW.NE.1) GO TO 311	IND01420
	RTARG=SQRT((PTS(4)-PTS(1))**2+(PTS(5)-PTS(2))**2+	IND01430
	+(PTS(6)-PTS(3))**2)	IND01440
	THETA=ACOS((PTS(3)-PTS(6))/RTARG)	IND01450
	RTDCON=1.0/PIRAD	IND01460
	THETA=THETA*RTDCON	IND01470
	ALT=PTS(6)	IND01480
	DELX=PTS(1)-PTS(4)	IND01490
	DELY=PTS(2)-PTS(5)	IND01500
	HDIS=SQRT(DELX**2+DELY**2)	IND01510
	PHI=ACOS(DELX/HDIS)	IND01520
	PHI=PHI*RTDCON	IND01530
	IF (DELY.LT.0.0) PHI=360.0-PHI	IND01540
311	CONTINUE	IND01550
	IF (ITARG.EQ.1) RTARG=ALT/ABS(COS(THETA*PIRAD))	IND01560
	IF (ITARG.EQ.0) RTARG=1000.0	IND01570
	RETURN	IND01580
400	CONTINUE	IND01590
	REWIND IPHFUN	IND01600
	DO 500 I=1,IWN	IND01610
500	PF(I)=0.	IND01620
	IF (NLAM.NE.0) CALL PFUNC(ID)	IND01630
	RETURN	IND01640
	END	IND01650


```

SUBROUTINE OUTPUT(MODEL, IHAZE, CLDHGT)
LOGICAL LMNLT
COMMON /MO1/ DUMMIE(715), MHOLD, NLHOLD, DUMMYS(1088),
+      TRANS(16,5), RADA(16,2), WAVE(16), SS(16),
1      DIR(16), RADG(16), UTEM(16), UTRF(16), BK(16),
2      PATHR(16), UERF(16), PATHR2(16), TOT(12), BKG(16)
COMMON /BKDAT/ ALT, THETA, PHI, SANG2, ZENTH, PHASE, ALB
COMMON /COMI1/ ISORC, ITARG, IWN, JHL
COMMON /ANSW2/ TTR(16), TBR(16), CNTRST(16)
COMMON /HOLRTH/ HITARG(8,3), HISORC(6,5),
1      HMODEL(5,6), HLIGHT(3), HMNLT(3), HSNLT(3),
2      HTRGT(2), HTYPE(2), HGRND(2)
COMMON /IOUNIT/ IOIN, IOOUT, IPHFUN, LOUNIT, NDIRTU, NCLIMT, KSTOR, NPLOTU
DATA HGRND/4H GRO,4HUND /
DATA HISORC/4H SUNL,4HIGHT,4H ONL,4HY,4H,4H,4H
1      4HMOON,4HLIGH,4HT ON,4HLY,4H,4H,4HEMIS,
2      4HSION,4H ONL,4HY,4H,4H,4HSUNL,4HIGHT,
3      4H AND,4H EMI,4HSSIO,4HN,4HMOON,4HLIGH,4HT AN,
4      4HD EM,4HISSI,4HON /
DATA HITARG/4HND R,4HEFLE,4HCTAN,4HCE,4H,4H,4H
1      4H,4H,4HGROU,4HND R,4HEFLE,4HCTAN,4HCE /,
2      4H EMI,4HSSIO,4HN,4HTARG,4HET R,4HEFLE,4HCTAN,
3      4HCE /,4H EMI,4HSSIO,4HN /
DATA HMODEL/4HTROP,4HICAL,4H,4H,4H,4HMIDA,
1      4HLTIT,4HUDE,4HSUMM,4HER,4HMIDA,4HLTIT,4HUDE,
2      4HWINT,4HER,4HSUBA,4HRCIT,4H SU,4HMMER,4H,
3      4HSUBA,4HRCIT,4HC WI,4HNTER,4H,4H1962,4H U.S,
4      4H, ST,4HANDA,4HRD /
DATA HMNLT/4H MO,4HONLI,4HGHT /
DATA HSNLT/4H SU,4HNLI,4HHT /
DATA HTRGT/4H TAR,4HGET /
DATA HTYPE/4H,4H /
DATA LMNLT/.FALSE./
IF (ISORC.NE.0.AND.ISORC.NE.3) GO TO 200
DO 100 I=1,3
100 HLIGHT(I)=HSNLT(I)
GO TO 400
DO 200 I=1,3
200 HLIGHT(I)=HMNLT(I)
LMNLT=.TRUE.
400 IF (ITARG.EQ.0) GO TO 600
IF (ITARG.EQ.2) GO TO 500
HTYPE(1)=HGRND(1)
HTYPE(2)=HGRND(2)
GO TO 600
500 HTYPE(1)=HTRGT(1)
HTYPE(2)=HTRGT(2)
600 CALL DIAG
IF (MODEL.GT.7) GO TO 700
WRITE (IOOUT,1400) ISORC,(HISORC(I,ISORC+1),I=1,6),
1      ITARG,(HITARG(I,ITARG+1),I=1,8),MODEL,
2      HMODEL(I,MODEL),I=1,5),IHAZE
GO TO 750
700 IF (MODEL.EQ.8) WRITE (IOOUT,1450) ISORC,(HISORC(I,ISORC+1),I=1,6),
1      ITARG,(HITARG(I,ITARG+1),I=1,8),MODEL,IHAZE
IF (MODEL.EQ.9) WRITE (IOOUT,1500) ISORC,(HISORC(I,ISORC+1),I=1,6),
1      ITARG,(HITARG(I,ITARG+1),I=1,8),MODEL,IHAZE
750 IF (IHAZE.GT.0) WRITE (IOOUT,1600) CLDHGT
800 WRITE (IOOUT,1700)
IF (LMNLT) WRITE (IOOUT,1800) PHASE
WRITE (IOOUT,1900) HLIGHT
DO 900 I=1,IWN
NW=10000./WAVE(I)
900 WRITE (IOOUT,2000) WAVE(I),NW,SS(I),BK(I),BKG(I)
WRITE (IOOUT,2100)
DO 1000 I=1,IWN
NW=10000./WAVE(I)
1000 WRITE (IOOUT,2200) WAVE(I),NW,UTEM(I),UTRF(I),
1      RADA(I,2),PATHR2(I),TTR(I)
WRITE (IOOUT,2300)

```

```

OUTPUT0010
OUTPUT0020
OUTPUT0030
OUTPUT0040
OUTPUT0050
OUTPUT0060
OUTPUT0070
OUTPUT0080
OUTPUT0090
OUTPUT0100
OUTPUT0110
OUTPUT0120
OUTPUT0130
OUTPUT0140
OUTPUT0150
OUTPUT0160
OUTPUT0170
OUTPUT0180
OUTPUT0190
OUTPUT0200
OUTPUT0210
OUTPUT0220
OUTPUT0230
OUTPUT0240
OUTPUT0250
OUTPUT0260
OUTPUT0270
OUTPUT0280
OUTPUT0290
OUTPUT0300
OUTPUT0310
OUTPUT0320
OUTPUT0330
OUTPUT0340
OUTPUT0350
OUTPUT0360
OUTPUT0370
OUTPUT0380
OUTPUT0390
OUTPUT0400
OUTPUT0410
OUTPUT0420
OUTPUT0430
OUTPUT0440
OUTPUT0450
OUTPUT0460
OUTPUT0470
OUTPUT0480
OUTPUT0490
OUTPUT0500
OUTPUT0510
OUTPUT0520
OUTPUT0530
OUTPUT0540
OUTPUT0550
OUTPUT0560
OUTPUT0570
OUTPUT0580
OUTPUT0590
OUTPUT0600
OUTPUT0610
OUTPUT0620
OUTPUT0630
OUTPUT0640
OUTPUT0650
OUTPUT0660
OUTPUT0670
OUTPUT0680
OUTPUT0690
OUTPUT0700

```

```

DO 1100 I=1, IWN
NW=10000./WAVE(I)
1100 WRITE (IOOUT,2200) WAVE(I),NW,RADG(I),UERF(I),
1 RADAC(I,1),PATHR(I),TBR(I)
WRITE (IOOUT,2400) HLIGHT
IF (LMNLT) WRITE (IOOUT,1800) PHASE
WRITE (IOOUT,2500) HLIGHT,HLIGHT
DO 1200 I=1, IWN
NW=10000./WAVE(I)
1200 WRITE (IOOUT,2600) WAVE(I),NW,SS(I),DIR(I)
WRITE (IOOUT,2700)
DO 1300 I=1, IWN
NW=10000./WAVE(I)
1300 WRITE (IOOUT,2800) WAVE(I),NW,TTR(I),TBR(I),CNTRST(I)
WRITE (IOOUT,2900) (TOT(I),I=1,10),TOT(12),HLIGHT,
1 TOT(11)
RETURN
C
1400 FORMAT (43X,37HDEFINITION OF CONTROL PARAMETERS
1 8HFOLLOWS:,,43X,10(1H-),6H -- ,7(1H-),2X,
2 10(1H-),2X,7(1H-),,,,43X,9HPARAMETER,3X,
3 5HVALUE,3X,11HDESCRIPTION,,43X,9(1H-),3X,5(
4 1H-),3X,11(1H-),,,45X,5HISORC,7X,11,5X,6A4,/,
5 45X,5HITARG,7X,11,5X,8A4,/,45X,5HMODEL,7X,
6 11,5X,5A4,/,45X,5HIHAZE,7X,11,/)
1450 FORMAT (43X,37HDEFINITION OF CONTROL PARAMETERS
1 8HFOLLOWS:,,43X,10(1H-),6H -- ,7(1H-),2X,
2 10(1H-),2X,7(1H-),,,,43X,9HPARAMETER,3X,
3 5HVALUE,3X,11HDESCRIPTION,,43X,9(1H-),3X,5(
4 1H-),3X,11(1H-),,,45X,5HISORC,7X,11,5X,6A4,/,
5 45X,5HITARG,7X,11,5X,8A4,/,45X,5HMODEL,7X,
6 11,5X,32HISRAELI STANDARD (YEAR, DAYTIME)/45X,5HIHAZE,7X,11/)
1500 FORMAT (43X,37HDEFINITION OF CONTROL PARAMETERS
1 8HFOLLOWS:,,43X,10(1H-),6H -- ,7(1H-),2X,
2 10(1H-),2X,7(1H-),,,,43X,9HPARAMETER,3X,
3 5HVALUE,3X,11HDESCRIPTION,,43X,9(1H-),3X,5(
4 1H-),3X,11(1H-),,,45X,5HISORC,7X,11,5X,6A4,/,
5 45X,5HITARG,7X,11,5X,8A4,/,45X,5HMODEL,7X,
6 11,5X,34HISRAELI STANDARD (YEAR, NIGHTTIME)/45X,5HIHAZE,7X,11/)
1600 FORMAT (1H,45X,22H CLOUD BOTTOM HEIGHT = ,F5.3,3H KM)
1700 FORMAT (1H1,56X,19HSOURCE INTENSITIES,,56X,7H-----
1 12H -----,/)
1800 FORMAT (44X,27H PHASE ANGLE FOR MOONLIGHT: ,F6.2,
1 10H (DEGREES),/)
1900 FORMAT (15X,10HWAVELENGTH,3X,10HWAVENUMBER,3X,3A4,
1 6HSOURCE,11X,13HTARGET SOURCE,15X,7HGROUND
2 6HSOURCE,,15X,9H(MICRONS),6X,6H(CM-1),11X,
3 8HSTRENGTH,17X,8HSTRENGTH,20X,8HSTRENGTH,/,
4 41X,20H(WATTS M-2 MICRON-1),2(3X,9H(WATTS M-
5 16H2 MICRON-1 SR-1)),/,15X,10(1H-),3X,10(1H-),
6 3X,20(1H-),3X,25(1H-),3X,25(1H-),/)
2000 FORMAT (15X,1PE10.4,3X,17,11X,1PE10.4,15X,1PE10.4,18X,
1 1PE10.4)
2100 FORMAT (1H1,46X,33HCOMPONENTS FOR RADIANCE FROM
1 6HTARGET,,46X,29H-----
2 10H-----,/,53X,23H(WATTS M-2 MICRON-1 SR-
3 2H1),/,22X,10HWAVELENGTH,3X,10HWAVENUMBER,
4 5X,6HTARGET,7X,6HTARGET,7X,7HPARTIAL,6X,
5 7HPARTIAL,6X,5HTOTAL,/,22X,9H(MICRONS),6X,
6 6H(CM-1),6X,8HEMISSION,4X,11HREFLECTANCE,2X,
7 11HATMOSPHERIC,5X,4HPATH,8X,6HTARGET,/,75X,
8 8HEMISSION,5X,8HRADIANCE,5X,8HRADIANCE,/,
9 22X,10(1H-),3X,10(1H-),4X,8(1H-),4X,11(1H-),
X 2X,11(1H-),3X,8(1H-),5X,8(1H-),/)
2200 FORMAT (22X,1PE10.4,3X,17,3X,1PE13.4)
2300 FORMAT (1H1,47X,29HCOMPONENTS FOR BACKGROUND
1 8HRADIANCE,,47X,27H-----
2 10H-----,/,53X,23H(WATTS M-2 MICRON-1 SR-
3 2H1),/,22X,10HWAVELENGTH,3X,10HWAVENUMBER,
4 5X,6HGROUND,7X,6HGROUND,8X,5HTOTAL,8X,5HTOTAL,

```

```

OUTP0710
OUTP0720
OUTP0730
OUTP0740
OUTP0750
OUTP0760
OUTP0770
OUTP0780
OUTP0790
OUTP0800
OUTP0810
OUTP0820
OUTP0830
OUTP0840
OUTP0850
OUTP0860
OUTP0870
OUTP0880
OUTP0890
OUTP0900
OUTP0910
OUTP0920
OUTP0930
OUTP0940
OUTP0950
OUTP0960
OUTP0970
OUTP0980
OUTP0990
OUTP1000
OUTP1010
OUTP1020
OUTP1030
OUTP1040
OUTP1050
OUTP1060
OUTP1070
OUTP1080
OUTP1090
OUTP1100
OUTP1110
OUTP1120
OUTP1130
OUTP1140
OUTP1150
OUTP1160
OUTP1170
OUTP1180
OUTP1190
OUTP1200
OUTP1210
OUTP1220
OUTP1230
OUTP1240
OUTP1250
OUTP1260
OUTP1270
OUTP1280
OUTP1290
OUTP1300
OUTP1310
OUTP1320
OUTP1330
OUTP1340
OUTP1350
OUTP1360
OUTP1370
OUTP1380
OUTP1390
OUTP1400

```

5	7X, 5HTOTAL, /, 22X, 9H(MICRONS), 6X, 6H(CM-1), 6X,	OUTP1410
6	8HEMISSION, 4X, 11HREFLECTANCE, 2X, 11HATMOSPHERIC	OUTP1420
7	, 5X, 4HPATH, 6X, 10HBACKGROUND, /, 75X, 8HEMISSION,	OUTP1430
8	2(5X, 8HRADIANCE), /, 22X, 10(1H-), 3X, 10(1H-), 4X,	OUTP1440
9	8(1H-), 4X, 11(1H-), 2X, 11(1H-), 3X, 8(1H-), 4X, 10(OUTP1450
X	1H-), /, /)	OUTP1460
2400	FORMAT (1H1, 58X, 6HDIRECT, 3A4, /, 57X, 6(1H-), 1X, 12(1H-), /	OUTP1470
1	, /, 56X, 20H(WATTS M-2 MICRON-1), /, /)	OUTP1480
2500	FORMAT (41X, 10HWAVELENGTH, 3X, 10HWAVENUMBER, 2X, 3A4, 1X,	OUTP1490
1	3A4, /, 41X, 9H(MICRONS), 6X, 6H(CM-1), 7X, 6HSOURCE,	OUTP1500
2	8X, 4HFLUX, /, 68X, 8HSTRENGTH, /, 41X, 10(1H-), 3X,	OUTP1510
3	10(1H-), 2X, 12(1H-), 1X, 12(1H-), /, /)	OUTP1520
2600	FORMAT (41X, 1PE10.4, 3X, 17, 6X, 1PE10.4, 3X, 1PE10.4)	OUTP1530
2700	FORMAT (1H1, 58X, 15HTOTAL RADIANCE, /, 58X, 11H-----	OUTP1540
1	4H-----, /, 53X, 25H(WATTS M-2 MICRON-1 SR-1), /, /)	OUTP1550
2	, 35X, 10HWAVELENGTH, 3X, 10HWAVENUMBER, 5X,	OUTP1560
3	6HTARGET, 5X, 10HBACKGROUND, 4X, 8HCONTRAST, /	OUTP1570
4	, 35X, 9H(MICRONS), 6X, 6H(CM-1), 33X, 5HRATIO, /	OUTP1580
5	, 35X, 10(1H-), 3X, 10(1H-), 5X, 6(1H-), 5X, 10(1H-),	OUTP1590
6	4X, 8(1H-), /, /)	OUTP1600
2800	FORMAT (35X, 1PE10.4, 3X, 17, 3X, 1P3E13.4)	OUTP1610
2900	FORMAT (1H1, 46X, 30HDETECTOR-RESPONSE WAVELENGTH-	OUTP1620
1	10HINTEGRATED, /, 46X, 17(1H-), 2X, 21(1H-), /, /	OUTP1630
2	, 58X, 16H(WATTS M-2 SR-1), /, /, 46X, 7HTARGET	OUTP1640
3	8HEMISSION, 16X, 1PE10.4, /, 46X, 7HTARGET	OUTP1650
4	11HREFLECTANCE, 13X, 1PE10.4, /, 46X, 8HPARTIAL	OUTP1660
5	20HATMOSPHERIC EMISSION, 3X, 1PE10.4, /, 46X,	OUTP1670
6	21HPARTIAL PATH RADIANCE, 10X, 1PE10.4, /, 46X,	OUTP1680
7	21HTOTAL TARGET RADIANCE, 10X, 1PE10.4, /, 46X,	OUTP1690
8	15HGROUND EMISSION, 16X, 1PE10.4, /, 46X, 7HGROUND	OUTP1700
9	11HREFLECTANCE, 13X, 1PE10.4, /, 46X, 6HTOTAL	OUTP1710
X	20HATMOSPHERIC EMISSION, 5X, 1PE10.4, /, 46X,	OUTP1720
1	19HTOTAL PATH RADIANCE, 12X, 1PE10.4, /, 46X,	OUTP1730
2	25HTOTAL BACKGROUND RADIANCE, 6X, 1PE10.4, /, /	OUTP1740
3	, 46X, 41(1H*), /, /, 46X, 8HCONTRAST, 22X, 1PE11.4, /, /	OUTP1750
4	, 46X, 41(1H*), /, /, 46X, 6HDIRECT, 3A4, 13X, 1PE10.4, /	OUTP1760
5	, 46X, 11H(WATTS M-2))	OUTP1770
	END	OUTP1780

```

SUBROUTINE PATHRD(CTHED,HP,RT,IO,IHAZE,NR,
1 IEMISS,LEN,MODEL,VIS,V1,V2,TGRD,DUMMY,ICLMAT,MULDV)
LOGICAL HORIZ
DIMENSION TR1(16),TR2(16),DUMMY(16),ANS(16)
COMMON /BASPT/ ANG(65),SUM(65),WVL(16),NWVL,ALBB(16),BS(16),
1 BE(16),SINGWV,SAER(65),LOUM
COMMON /BKDAT/ALT,THETA,PHI,SANG2,ZENTH,PHASE,ALB
COMMON /CGEOM/COSGM,COSAT,CSA
COMMON /COM11/ISORC,ITARG,IWN,JHL
COMMON /MQ1/EH(16,34),P(34),T(34),WH(34),Z(34),WA(34),RE,M,NL,
+ RRS(16,34),SCOE(16,34),
+ TRANS(16,5),RADA(16,2),WAVE(16),SS(16),
1 DIR(16),RADG(16),UTEM(16),UTRF(16),BK(16),
2 PATHR(16),UERF(16),PATHR2(16),TOT(12),BKG(16)
C ***** CONST1 = (1.-.0295)*.5*3.14159/(1+.5*.0295)
C ***** CONST2 = .0295/(1+.5*.0295)*3.14159
DATA CONST1,CONST2/.0570805145,.0034701189/
NLL=NL-1
SRAYL=CONST1*(1.+CSA*CSA)+CONST2
C ***** INITIALIZE VARIABLES *****
DO 800 IW=1,IWN
800 PATHR(IW)=0.0
DO 900 J=2,NLL
900 IF (HP.LT.Z(J)) GO TO 1000
CONTINUE
J=NLL
JU=J
JL=JU-1
HORIZ=.FALSE.
DS1=0.0
H2=HP
RAT=(H2-Z(JL))/(Z(JU)-Z(JL))
GO TO (1100,1200,1300),IO
1100 IX=RT*1.99999+1
DX=RT/FLOAT(IX)
DH=DX*CTHED
HORIZ=ABS(CTHED).LT.1.E-3
DIST=0.5*DX
GO TO 1400
1200 IX=NLL-JL
DH=Z(JU)-HP
DX=DH/CTHED
GO TO 1400
C HORIZONTAL PATH FOLLOWS
1300 IX=100
DIST=DIST-0.5*DX+0.25
DX=0.5
HORIZ=.TRUE.
GO TO 1500
1400 H2=HP+DH*0.5
1500 DO 2600 K=1,IX
IF (HORIZ) GO TO 2000
CALL LT4M(HP,H2,THETA,2,2,TR1,DUMMY,DUMMY,
1 IEMISS,LEN,MODEL,VIS,V1,V2,TGRD,ICLMAT,IERR,
2 NR,IHAZE,MULDV)
CALL LT4M(H2,LOUM,ZENTH,3,2,TR2,DUMMY,DUMMY,
1 IEMISS,LEN,MODEL,VIS,V1,V2,TGRD,ICLMAT,IERR,
2 NR,IHAZE,MULDV)
IF (IO.EQ.2) GO TO 1800
DO 1600 J=2,NLL
IF (H2.LT.Z(J)) GO TO 1700
1600 CONTINUE
J=NLL
1700 JU=J
JL=JU-1
RAT=(H2-Z(JL))/(Z(JU)-Z(JL))
DO 1800 IW=1,IWN
SC=SCOE(IW,JL)+RAT*(SCOE(IW,JU)-SCOE(IW,JL))
RS=RRS(IW,JL)+RAT*(RRS(IW,JU)-RRS(IW,JL))
SSCAT=RS*SRAYL+(1.0-RS)*SAER(IW)

```

```

PAD00010
PAD00020
PAD00030
PAD00040
PAD00050
PAD00060
PAD00070
PAD00080
PAD00090
PAD00100
PAD00110
PAD00120
PAD00130
PAD00140
PAD00150
PAD00160
PAD00170
PAD00180
PAD00190
PAD00200
PAD00210
PAD00220
PAD00230
PAD00240
PAD00250
PAD00260
PAD00270
PAD00280
PAD00290
PAD00300
PAD00310
PAD00320
PAD00330
PAD00340
PAD00350
PAD00360
PAD00370
PAD00380
PAD00390
PAD00400
PAD00410
PAD00420
PAD00430
PAD00440
PAD00450
PAD00460
PAD00470
PAD00480
PAD00490
PAD00500
PAD00510
PAD00520
PAD00530
PAD00540
PAD00550
PAD00560
PAD00570
PAD00580
PAD00590
PAD00600
PAD00610
PAD00620
PAD00630
PAD00640
PAD00650
PAD00660
PAD00670
PAD00680
PAD00690
PAD00700

```

1900	PATHR(IW)=PATHR(IW)+TR1(IW)*TR2(IW)*SC*SSCAT*DX	PAD00710
	GO TO (2400,2500,2300),IO	PAD00720
2000	CONTINUE	PAD00730
	CALL LT4M(HP,DUM,DIST,1,2,TR1,DUMMY,DUMMY,	PAD00740
1	IEMISS,LEN,MODEL,VIS, V1,V2,TGRD,ICLMAT,IERR,	PAD00750
2	NR,HAZE,MULDV)	PAD00760
	IF (K.EQ.1) CALL LT4M(HP,DUM,ZENTH,3,2,TR2,DUMMY,DUMMY,	PAD00770
1	IEMISS,LEN,MODEL,VIS, V1,V2,TGRD,	PAD00780
2	ICLMAT,IERR,	PAD00790
2	NR,HAZE,MULDV)	PAD00800
	DS=0.0	PAD00810
	DO 2200 IW=1,IWN	PAD00820
	IF (K.GT.1) GO TO 2100	PAD00830
	SC=SCOE(IW,JL)+RAT*(SCOE(IW,JU)-SCOE(IW,JL))	PAD00840
	RS=RRS(IW,JL)+RAT*(RRS(IW,JU)-RRS(IW,JL))	PAD00850
	ANS(IW)=TR2(IW)*SC*(RS*SRAYL+(1.0-RS)*SAER(IW))	PAD00860
2100	DPATH=ANS(IW)*TR1(IW)*DX	PAD00870
	DS=DS+DPATH	PAD00880
2200	PATHR(IW)=PATHR(IW)+DPATH	PAD00890
	DS1=DS1+DS	PAD00900
	DS=DS*0.5/DX	PAD00910
	IF (IO.EQ.3.AND.K.GT.1.AND.DS/DS1.LT.0.001) RETURN	PAD00920
	IK=(K/20)*20	PAD00930
	IF (IO.EQ.3.AND.IK.EQ.K) DX=DX*2.0	PAD00940
2300	DIST=DIST+DX	PAD00950
	GO TO 2600	PAD00960
2400	H2=H2+DH	PAD00970
	GO TO 2600	PAD00980
2500	JU=JU+1	PAD00990
	JL=JL+1	PAD01000
	IF (K.EQ.1) RAT=0.5	PAD01010
	H2=(Z(JU)+Z(JL))*0.5	PAD01020
	DX=(Z(JU)-Z(JL))/CTHED	PAD01030
2600	CONTINUE	PAD01040
	RETURN	PAD01050
	END	PAD01060

```
SUBROUTINE ZERO  
COMMON /MO1/ DUMMIE(715),M,NL,DUMMYS(1088),RDATA(300)  
DO 100 I=1,300  
RDATA(I)=0.0  
100 CONTINUE  
RETURN  
END
```

```
ZER00010  
ZER00020  
ZER00030  
ZER00040  
ZER00050  
ZER00060  
ZER00070
```

SUBROUTINE CLIMAT(LOCAT, MONTH, NHOUR, IWIND, NPRT, TEMP, PRESS, RH, AH, DP, CL100010
1, VIS, WNDVEL, WINDIR, IPASCT) CL100020

***** CL100030

CLIMATOLOGY MODULE - CLIMAT

PURPOSE - TO PROVIDE THE CENTRAL EUROPEAN AND MID-EASTERN CLIMATOLOGY DATA REQUIRED BY OTHER MODULES OF EOSAEL. CL100040
CL100050

PARAMETER DESCRIPTION

LOCAT - CLIMATOLOGY REGION INDICATOR. LOCAT IS AN INTEGER
(1-4) FOR CENTRAL EUROPE AND
(5-10) FOR MID-EAST. CL100060
CL100070
CL100080
CL100090
CL100100
CL100110
CL100120
CL100130
CL100140
CL100150

REGION 1 - EUROPEAN LOWLANDS,
REGION 2 - EUROPEAN RHINE VALLEY,
REGION 3 - EUROPEAN HIGHLANDS,
REGION 4 - EUROPEAN ALPINE,
REGION 5 - MIDEAST DESERTS,
REGION 6 - MIDEAST COASTAL,
REGION 7 - MIDEAST PERSIAN GULF,
REGION 8 - MIDEAST RED SEA,
REGION 9 - MIDEAST EASTERN MOUNTAINS, AND
REGION 10 - MIDEAST INDUS VALLEY. CL100160
CL100170
CL100180
CL100190
CL100200
CL100210
CL100220
CL100230
CL100240
CL100250
CL100260

MONTH - AN INTEGER (1-12) INDICATING THE MONTH OF THE YEAR.
MONTH IS USED TO SELECT THE SEASON WHICH IS
APPLICABLE TO THE REGION LOCAT. CL100270
CL100280

NHOUR - AN INTEGER (0-23) INDICATING THE TIME OF DAY LOCAL
STANDARD TIME (LST). NHOUR IS USED TO SELECT ONE OF
FOUR TIME PERIODS OF THE DAY 20-02, 03-09, 10-14,
AND 15-19. CL100290
CL100300
CL100310
CL100320
CL100330

IWIND - *** NOT USED *** CL100340

NPRT - A PRINT SELECTOR. CL100350

NPRT LE ZERO - DO NOT PRINT CLIMATOLOGICAL DATA.
NPRT GT ZERO - PRINT ALL AVAILABLE MEANS, STANDARD
DEVIATIONS, AND PERCENT OCCURRENCES. CL100360
CL100370
CL100380
CL100390

TEMP - MEAN TEMPERATURE (C). CL100400

PRESS - MEAN SEA LEVEL PRESSURE (MB). CL100410

RH - MEAN RELATIVE HUMIDITY (PERCENT). CL100420

AH - MEAN ABSOLUTE HUMIDITY (GM/CM³). CL100430

DP - MEAN DEW-POINT TEMPERATURE (C). CL100440

VIS - MEAN HORIZONTAL VISIBILITY (KM). CL100450

WNDVEL - MEAN WIND SPEED (MPS). CL100460

WINDIR - MOST PROBABLE WIND DIRECTION (DEGREES). WINDIR IS
GIVEN IN 30 DEGREE INCREMENTS (015, 045, 075, ...
345). CL100470
CL100480

IPASCT - INDICATOR (1-6) FOR THE MOST PROBABLE PASQUILL
STABILITY CATEGORY (A-F). CL100490
CL100500

CLDHT - MEAN CLOUD HEIGHT (KM). CL100510

CLDCVR - MEAN TOTAL CLOUD COVER (PERCENT). CL100520

WINDIR - WIND DIRECTION (DEGREES). CL100530
CL100540
CL100550

SUBROUTINES AND FUNCTIONS - NONE CL100560

CARD INPUT - NONE CL100570
CL100580

TAPE INPUT - YES. BE SURE TO ASSIGN THE CLIMATOLOGY DATA TAPE TO
UNIT NCLINT. CL100590
CL100600
CL100610

***** CL100620

COMMON /IOUNIT/IOIN, IOOUT, IPHFUN, LOUNIT, NDIRTU, NCLINT, KSTOR, NPLOTU, CL100630
DIMENSION REGION(70), SEASON(8), HOUR(8), DATA(18), DIR(13) CL100640
CL100650
CL100660
CL100670

DATA REGION/4HEURO, 4HPEAN, 4H LOW, 4HLAND, 4HS
1 2*4H , 4HEURO, 4HPEAN, 4H RHI, 4HNE V, 4HALL, 4HY , 4H , 4HEURO, CL100680
2 4HPEAN, 4H HIG, 4HHLAN, 4HDS , 2*4H , 4HEURO, 4HPEAN, 4H ALP, 4HINE, CL100690
3 3*4H , 4HMIDE, 4HAST , 4HDESE, 4HRTS , 3*4H , 4HMIDE, 4HAST , 4HCOAC CL100700

```

4S,4HTAL,3*4H,4HMIDE,4HAST,4HPERS,4HIAN,4HGULF,2*4H,4HMICLI00710
5DE,4HAST,4HRED,4HSEA,3*4H,4HMIDE,4HAST,4HEAST,4HERN,4HMOUCL00720
6N,4HTAIN,4HS,4HMIDE,4HAST,4HINDU,4HS VA,4HLEY,2*4H,CL00730
DATA SEASON/4HWINT,4HER,4HSPRI,4HNG,4HSUMM,4HER,4HAUTU,4HMN,CL00740
1,CL00750
DATA HOUR/4H20-0,4H2,4H03-0,4H9,4H10-1,4H4,4H15-1,4H9,CL00760
DATA DIR/4H 015,4H 045,4H 075,4H 105,4H 135,4H 165,4H 195,4H 225,CL00770
14H 255,4H 285,4H 315,4H 345,4H VBL,CL00780
POSITION THE TAPE NCLINT FOR READING,CL00790
REWIND NCLINT,CL00800
SKIP OVER ALL DATA FOR REGIONS 1,2,...,LOCAT-1,CL00810
IF(LOCAT.LT.1.OR.LOCAT.GT.10) LOCAT=1,CL00820
LSKIP=1056*(LOCAT-1),CL00830
SKIP OVER DATA FOR SEASONS 1,2,...,SEASON-1 FOR REGION LOCAT,CL00840
NSEASN=1,CL00850
IF(MONTH.GE.3.AND.MONTH.LE.5) NSEASN=2,CL00860
IF(MONTH.GE.6.AND.MONTH.LE.8) NSEASN=3,CL00870
IF(MONTH.GE.9.AND.MONTH.LE.10) NSEASN=4,CL00880
IF(LOCAT.GE.5.AND.MONTH.EQ.11) NSEASN=4,CL00890
NSKIP=LSKIP+176*(NSEASN-1),CL00900
SKIP OVER DATA FOR TIME PERIODS 0,1,...,PERIOD-1 FOR REGION,CL00910
LOCAT DURING SEASON,CL00920
NTIME=1,CL00930
IF(NHOUR.GE.3.AND.NHOUR.LE.9) NTIME=2,CL00940
IF(NHOUR.GE.10.AND.NHOUR.LE.14) NTIME=3,CL00950
IF(NHOUR.GE.15.AND.NHOUR.LE.19) NTIME=4,CL00960
NSKIP=NSKIP+44*(NTIME-1),CL00970
IF(NSKIP.LE.0) GO TO 2,CL00980
DO 1 J=1,NSKIP,CL00990
READ(NCLINT,9) A,CL01000
1 CONTINUE,CL01010
IF NPRT GT 0, PRINT A HEADING FOR THE THERMODYNAMIC DATA,CL01020
2 CONTINUE,CL01030
IF(NPRT.LE.0) GO TO 3,CL01040
WRITE(IOOUT,15),CL01050
ILOC=7*LOCAT-6,CL01060
ILO6=ILOC+6,CL01070
WRITE(IOOUT,10) (REGION(J),J=ILOC,ILOC6),SEASON(2*NSEASN-1),CL01080
1 SEASON(2*NSEASN),HOUR(2*NTIME-1),HOUR(2*NTIME),CL01090
READ THE THERMODYNAMIC DATA FOR REGION LOCAT AT NTIME,CL01100
DURING NSEASN,CL01110
3 DO 4 J=1,22,CL01120
READ(NCLINT,11) NCLASS,(DATA(K),K=1,18),CL01130
CONVERT FROM METERS TO KILOMETERS,CL01140
DATA(6)=0.001*DATA(6),CL01150
DATA(10)=0.001*DATA(10),CL01160
IF NPRT GT 0, PRINT THE THERMODYNAMIC DATA,CL01170
IF(NPRT.LE.0) GO TO 4,CL01180
WRITE(IOOUT,12) NCLASS,(DATA(K),K=1,18),CL01190
4 CONTINUE,CL01200
EXTRACT THE VALUES OF TEMP, PRESS, RH, AH, DP, VIS, AND,CL01210
WINDVEL,CL01220
CL01230
CL01240
CL01250
CL01260
CL01270
CL01280
CL01290
CL01300
CL01310
CL01320
CL01330
CL01340
CL01350
CL01360
CL01370
CL01380
CL01390
CL01400

```



```

TEMP=DATA(2)
DP=DATA(3)
AH=DATA(4)
RH=DATA(5)
VIS=DATA(6)
PRESS=DATA(7)
WINDVEL=DATA(8)
DCC
    DETERMINE THE VALUE OF IPASCT
        IPASCT=1
        FREQ=DATA(13)
        DO 5 J=2,6
            IF(DATA(J+12),LE,FREQ) GO TO 5
            FREQ=DATA(J+12)
            IPASCT=J
5 CONTINUE
        GET TO THE WIND DATA ON NCLIMT
            NSKIP=(NSKIP-LSKIP)/44
            NSKIP=44*(15-NSKIP)+22*NSKIP
            DO 16 J=1,NSKIP
                READ(NCLIMT,9) A
16 CONTINUE
            IF NPRT GT 0, PRINT A HEADING FOR THE WIND DATA
            IF(NPRT.LE.0) GO TO 6
            WRITE(IOOUT,13) (DIR(J),J=1,13)
            READ THE WIND DATA FOR REGION LOCAT AT NTIME DURING NSEASN
6 DO 7 J=1,22
    READ(NCLIMT,14) NCLASS,(DATA(K),K=1,14)
        IF NPRT GT 0, PRINT THE WIND DATA
        IF(NPRT.LE.0) GO TO 7
        WRITE(IOOUT,17) NCLASS,(DATA(K),K=1,14)
7 CONTINUE
        DETERMINE THE VALUE OF WINDIR
            NDIR=1
            FREQ=DATA(2)
            DO 8 J=2,12
                IF(DATA(J+1),LE,FREQ) GO TO 8
                FREQ=DATA(J+1)
            NDIR=J
8 CONTINUE
            WINDIR=30*NDIR-15
            RETURN FROM CLIMAT
RETURN
FORMAT STATEMENTS
9 FORMAT(A1)
10 FORMAT(25H1 EOSAEL CLIMATOLOGY FOR ,7A4,8H DURING ,2A4,4H AT ,2A4,
17HCLST). //126H CLASS   FREQCY   MEAN   MEAN   MEAN   MEAN
2 MEAN MEAN/STDEV MEAN   MEAN/STDEV FREQCY FREQCY FREQCY FREQCY
3CY FREQCY/126H NO.     CLASS   TEMP   DP   AH   RH   VIS   PCL
4RESS   WNDVEL   CLDHT   CLDCVR   A       B       C       D       E
5      F /126H          (%)      (C)      (C)      (GM/CU,M) (%)      (KM)      (
6MB)      (MPS)      (KM)      (%)      (%)      (%)      (%)      (%)      (%)
7      (%) //
11 FORMAT(6X,I3,5F5.1,F7.0,F6.1,2F5.1,F6.0/8F5.1)
12 FORMAT(I5,F9.1,4F7.1,F7.3,F7.1,F4.1,1H/,F4.1,F7.3,F6.1,1H/,F5.1,6F

```

```

17.1)
13 FORMAT(1H0,7H CLASS ,14(7H FREQCY)/14H NO. CLASS,13(7H WND DIR) CL102110
1/10X,3H(%) ,1X,13(2X,A4,1X)/14X,13(3X,3H(%) ,1X)/ CL102120
14 FORMAT(6X,13,14F5.1) CL102130
15 FORMAT(18H1CLIMATOLOGY MODEL///39H DEFINITIONS OF METEOROLOGICAL CL102140
1CLASSES//48H CLASS 1 = FOG, HAZE AND MIST WITH VIS LT 1 KM./53H CL102150
2CLASS 2 = FOG, HAZE AND MIST WITH 1 LE VIS LT 3 KM./53H CLASS 3 =CL102160
3 FOG, HAZE AND MIST WITH 3 LE VIS LT 7 KM./48H CLASS 4 = FOG, HAZCL102170
4E AND MIST WITH VIS GE 7 KM./34H CLASS 5 = DUST WITH VIS LT 3 KM.CL102180
5/34H CLASS 6 = DUST WITH VIS GE 3 KM./53H CLASS 7 = DRIZZLE, RAICL102190
6N AND TSTMS WITH VIS LT 1 KM./58H CLASS 8 = DRIZZLE, RAIN AND TSTCL102200
7MS WITH 1 LE VIS LT 3 KM./58H CLASS 9 = DRIZZLE, RAIN AND TSTMS WCL102210
8ITH 3 LE VIS LT 7 KM./53H CLASS 10 = DRIZZLE, RAIN AND TSTMS WITH CL102220
9VIS GE 7 KM./34H CLASS 11 = SNOW WITH VIS LT 1 KM./39H CLASS 12 = CL102230
ASNOW WITH 1 LE VIS LT 3 KM./39H CLASS 13 = SNOW WITH 3 LE VIS LT 7CL102240
B KM./34H CLASS 14 = SNOW WITH VIS GE 7 KM./59H CLASS 15 = NO WEATHCL102250
CER AND ABSOLUTE HUMIDITY LT 10 GM/CU M./59H CLASS 16 = NO WEATHER CL102260
DAND ABSOLUTE HUMIDITY GE 10 GM/CU M./52H CLASS 17 = VIS LT 1 KM ANCL102270
ED CEILING HEIGHT LT 300 M./53H CLASS 18 = VIS LT 3 KM AND CEILING CL102280
FHEIGHT LT 1000 M./36H CLASS 19 = CEILING HEIGHT LT 300 M./37H CLASCL102290
GS 20 = CEILING HEIGHT LT 1000 M./23H CLASS 21 = NO CEILING./36H CLCL102300
HASS 22 = ALL CONDITIONS COMBINED.) CL102310
17 FORMAT(15,F9.1,13F7.1) CL102320
C CL102330
END CL102340
CL102350

```

PROGRAM AGAUS

PROGRAM AGAUS

REVISION DATE 22 JANUARY, 1982

PURPOSE:

TO CALCULATE EXTINCTION COEFFICIENTS, ETC., AND PRODUCE
LEGENDRE EXPANSION COEFFICIENTS, PHASE FUNCTIONS AND (OPTIONALLY)
SCATTERING FRACTIONS UNDER A VARIETY OF CONDITIONS AND AEROSOL
DISTRIBUTIONS AT ONE OR MORE WAVELENGTHS. THE PHASE FUNCTION
IS NORMALIZED TO 4 PI OMEGA ZERO AND MAY BE RENORMALIZED BY
DIVISION BY THE APPROPRIATE CONSTANT(S).

***** INPUT *****

CARD 1 - IDENTIFIER - 80 ALPHA CHARACTERS

CARD 2 - INTEGER CONTROL PARAMETERS: NWAVE, NINDX, IW, IDSTP,
NRADI, IT, MQRTE, IANG, IEO, NEOU
FORMAT (10I5)

NWAVE: IS THE NO. OF WAVELENGTHS, OR REL. HUMIDITY VALUES TO BE
TREATED IN THIS RUN. SEE COMMENTS CIRCA READ OF WAVE, DWAVE, ETC.
N.B. NWAVE MUST BE .LE. 10 - TO CHANGE THE NUMBER OF WAVELENGTHS
CHANGE THE FIRST INDEX OF ARRAY OUT(I,J) TO AGREE WITH NWAVE.
NINDX: IS THE NBR OF AEROSOL COMPONENTS WHICH WILL HAVE DIFFERENT
OPTICAL CONSTANTS, MASS DENSITIES OR MASS CONCENTRATIONS.
IW: =0 WILL SET THE REFRACTIVE INDEX OF THE AEROSOL EQUAL
TO THAT OF WATER AT THE INPUT WAVELENGTH AND TEMP. IF IW .NE. 0
AND HANEL'S GROWTH FACTOR IS ZERO (EMUA=0. - CARD 5), THEN THE
INPUT REFRACTIVE INDEX (EMA,CAYA) WILL BE USED FOR THE AEROSOL.
OTHERWISE THE REFRACTIVE INDEX IS ADJUSTED PER HANEL (SEE BELOW).
IDSTP: IDENTIFIES TYPE OF AEROSOL SIZE DISTRIBUTION TO BE USED.
NRADI: NO. OF PARTICLE RADII TO BE EXPECTED FOR IDSTP=0 OR 3;
THE INPUT VALUE OF NRADI IS IGNORED FOR IDSTP NOT ZERO OR 3.
NRADI MUST BE .LE. 1+2**JOIMCK(2) - C.F. BLOCK DATA
IT: IS THE NUMBER OF GAUSS-LEGENDRE ANGLES (ORDER OF EXPANSION)
IF ONLY EXTINCTION COEFFICIENTS, ETC. ARE DESIRED, I.E. NOT PHASE
FUNCTIONS, THEN SET -IT- EQUAL TO ONE.
MQRTE:=12345 WILL CAUSE PRINTS OF MIE EFFICIENCY FACTORS AT
EVERY VALUE OF PARTICLE RADIUS USED IN THE MIE CALCULATIONS;
SET MQRTE = 0 IF SUCH PRINTS ARE NOT DESIRED.
IANG:=0 FOR COMPUTATIONS OF PHASE FN. AT -IT- GAUSS LEGENDRE
QUADRATURE ANGLES; IANG=1 FOR COMPUTATIONS OF PHASE FN AT
-IT- EQUALLY SPACED ANGLES BETWEEN 0 AND 180 DEGREES.
IANG=2 WILL ALLOW -IT- USER SUPPLIED ANGLES TO BE READ -
FORMAT (16F5.1). THIS REQUIRES AT LEAST ONE CARD OF TYPE 2A.
IF IANG.GT.0 NO LEGENDRE COEFFICIENTS WILL BE GENERATED.
IEO=1,2,3,4 WILL CONSTRUCT A PHASE FUNCTION FILE (ON NEOU).
IEO=1 65 PREDETERMINED ANGLES INDIVIDUAL WAVELENGTHS ONLY
IEO=2 65 PREDETERMINED ANGLES COMPOSITE WAVELENGTH ONLY
IEO=3 USER INPUT ANGLES INDIVIDUAL WAVELENGTHS ONLY
IEO=4 USER INPUT ANGLES COMPOSITE WAVELENGTH ONLY
IEO=5 65 PREDETERMINED ANGLES INDIVIDUAL & COMPOSITE WAVELENGTHS
THE COMPOSITE WILL BE THE LAST DATA SET WRITTEN ON UNIT -NEOU-

FOR USER INPUT ANGLES SEE -IT- AND IANG ABOVE. THE COMPOSITE
VALUES ARE SIMPLE AVERAGES OVER THE NUMBER OF WAVELENGTHS.
THIS FILE WILL CONTAIN THE FOLLOWING INFORMATION:
1) ANGLES (65 MAX) - FORMAT(11(F6.2,1X))
2) NBR OF ANGLES, PHASE FUNCTION IDENTIFIER (=0= IMPLIED USER INPUT
IN EOSAEL), WAVELENGTH (UM), SINGLE SCATTERING ALBEDO, EXTINCTION
AND SCATTERING COEFFICIENTS IN INVERSE KM -
FORMAT (2(I2,1X),F5.2,1X,F8.6,1X,2(E12.6,1X))
3) PHASE FUNCTION AT ANGLES SPECIFIED ABOVE. N.B. THE PHASE FUNCTION
AS WRITTEN OUT HERE IS NORMALIZED TO 4 PI OMEGA ZERO; THE ROUTINE
IN EOSAEL WILL RENORMALIZE THE PHASE FUNCTION TO ONE.
FORMAT (6(E12.6,1X))
NEOU= UNIT NUMBER UPON WHICH EOSAEL PHASE FUNCTION IS TO BE STORED.

AGX00020
AGX00030
AGX00040
AGX00050
AGX00060
AGX00070
AGX00080
AGX00090
AGX00100
AGX00110
AGX00120
AGX00130
AGX00140
AGX00150
AGX00160
AGX00170
AGX00180
AGX00190
AGX00200
AGX00210
AGX00220
AGX00230
AGX00240
AGX00250
AGX00260
AGX00270
AGX00280
AGX00290
AGX00300
AGX00310
AGX00320
AGX00330
AGX00340
AGX00350
AGX00360
AGX00370
AGX00380
AGX00390
AGX00400
AGX00410
AGX00420
AGX00430
AGX00440
AGX00450
AGX00460
AGX00470
AGX00480
AGX00490
AGX00500
AGX00510
AGX00520
AGX00530
AGX00540
AGX00550
AGX00560
AGX00570
AGX00580
AGX00590
AGX00600
AGX00610
AGX00620
AGX00630
AGX00640
AGX00650
AGX00660
AGX00670
AGX00680
AGX00690
AGX00700

CARD 2A - USER SUPPLIED SET OF -IT- ANGLES. FORMAT (16F5.1)
16 VALUES PER CARD, MORE THAN 1 CARD MAY BE NEEDED.
THIS CARD IS ONLY NEEDED WHEN IANG=2.

AGX00710
AGX00720
AGX00730
AGX00740
AGX00750
AGX00760
AGX00770

CARD 3 - DISTRIBUTION PARAMETERS; ONLY ONE TYPE PER RUN.

FORMAT (6E12.6) - READ IN AGXP1

***** ALL DIMENSIONS ARE IN MICRONS *****

TYPE 0. USER SUPPLIED - NRADI CARDS, ONE VALUE OF RADIUS AND
NUMBER DENSITY PER CARD. IT IS SUGGESTED THAT DELTA BE
INPUT NO GREATER THAN .001 IN ORDER TO FORCE THE MAXIMUM
NUMBER OF RADII TO BE USED DUE TO THE POSSIBLE IRREGULAR
NATURE OF THIS DISTRIBUTION.
R(I), FF(I), I = 1, NRADI

TYPE 1. LOG-NORMAL
RBAR, SIGMA, RLO, RHI

TYPE 2. DOUBLE EXPONENTIAL

RLO, RHI, CUE, A, B

TYPE 3. DEIRMENDJIAN MODEL C

- NO INPUT -

TYPE 4. POWER LAW (JUNGE)

RLO, RHI, CUE, A

TYPE 5. MODIFIED GAMMA

RLO, RHI, RC, ALF, GAM

TYPE 6. MODIFIED GAMMA FOG MODEL

RLO, RHI, RC, ALF, GAM, ELWC

TYPE 7. POWER LAW

VIS

TYPE 8. CONTINENTAL BIMODAL

- NO INPUT -

TYPE 9. MARITIME BIMODAL

- NO INPUT -

TYPE 10. URBAN BIMODAL

- NO INPUT -

TYPE 11. USER SUPPLIED BIMODAL

FOA, RBARA, SGA, FOC, RBARC, SGC

TYPE 12. MARSHALL-PALMER RAIN MODEL

RAIN

AGX00810
AGX00820
AGX00830
AGX00840
AGX00850
AGX00860
AGX00870
AGX00880
AGX00890
AGX00900
AGX00910
AGX00920
AGX00930
AGX00940
AGX00950
AGX00960
AGX00970
AGX00980
AGX00990
AGX01000
AGX01010
AGX01020
AGX01030
AGX01040
AGX01050
AGX01060
AGX01070
AGX01080
AGX01090
AGX01100
AGX01110
AGX01120
AGX01130
AGX01140
AGX01150
AGX01160
AGX01170
AGX01180
AGX01190
AGX01200
AGX01210
AGX01220
AGX01230
AGX01240
AGX01250
AGX01260
AGX01270
AGX01280
AGX01290
AGX01300
AGX01310
AGX01320
AGX01330
AGX01340
AGX01350
AGX01360
AGX01370

CARD 4 - CONTROL PARAMETERS; FORMAT (6E12.6)

WAVE, DWAVE, RELHUM, DENS, TEMP, DELTA

FOR LOOPING OVER RELATIVE HUMIDITY ADD

NWAVE-1 CARDS CONTAINING RELHUM,TEMP - FORMAT(2E12.6)

SEE DWAVE BELOW.

WAVE: IS WAVELENGTH IN MICROMETERS.

DWAVE: IS THE WAVELENGTH INCREMENT IN MICROMETERS. IF DWAVE IS

LESS THAN 1E-4, A SPECIAL CASE APPLIES USED FOR LOOPING OVER

NWAVE VALUES OF RELHUM: THE FIRST TIME THIS CARD IS READ IT

MUST CONTAIN WAVE,DWAVE,RELHUM,DENS,TEMP,DELTA; THE SECOND AND

SUBSEQUENT TIMES IT MUST ONLY HAVE RELHUM,TEMP ON IT. THIS

ALSO REQUIRES REPETITION OF CARD 5.

RELHUM: IS RELATIVE HUMIDITY IN PERCENT.

DENS: IS PARTICLE NUMBER PER CUBIC CENTIMETER.

USER-SUPPLIED VALUE OF DENS WILL BE IGNORED FOR IDSTP=3 OR GT 6

BECAUSE THOSE DISTRIBUTIONS CARRY PRE-DETERMINED DENSITY VALUES.

ALSO, IF DENS IS LESS THAN 1E-4, THE PARTICLE NUMBER DENSITY

WILL BE CALCULATED FROM MASS DENSITY AND MASS CONCENTRATION.

TEMP: IS THE TEMPERATURE OF THE ATMOSPHERE IN DEGREES C.

DELTA: IS THE CONVERGENCE CRITERION WITHIN A PARTICULAR SIZE

RANGE INTERVAL; HALVING IS TERMINATED WHEN THE QUANTITY DEL

IS LESS THAN DELTA. **N.B.** THE AMOUNT OF CPU TIME USED BY

THIS PROGRAM IS CLOSELY CONNECTED WITH DELTA. THE SMALLER DELTA

IS THE LARGER YOUR RUN TIME WILL BE. IT IS SUGGESTED THAT

DELTA BE SET EQUAL TO .001 FOR MOST RUNS.

CARD 5 - OPTICAL AND PHYSICAL DATA; FORMAT (4F10.5,E15.7)

EMA, CAYA, EMUA, RHOA, CONC

REPEAT NWAVE*NINDX TIMES; IF IDSTP=6 THIS CARD IS NOT NEEDED.

EMA: IS THE REAL PART OF THE INDEX OF REFRACTION OF DRY AEROSOL.

CAYA: IS THE IMAGINARY PART OF REFRACTIVE INDEX FOR DRY AEROSOL.

***** CAYA IS ASSUMED TO BE NEGATIVE *****

```

C      **** DO NOT ENTER CAYA WITH A NEGATIVE SIGN !!!!! ****
C      EMUA: IS HANEL'S GROWTH FACTOR (MU-BAR)/ACCRETION COEF.
C      RHOA: IS THE MASS DENSITY (SP. GRAV) OF DRY AEROSOL.
C      CONC: IS THE MASS CONCENTRATION (GM/CC) OF DRY AEROSOL.
C
C      ***** END INPUT *****
C
C      ***** MISCELLANEOUS INFO *****
C
C      THE INPUT AND OUTPUT UNITS, ALONG WITH A EXTRA, CURRENTLY UNUSED,
C      UNIT (NUNIT) ARE ASSIGNED VALUES IN THE BLOCK DATA SUBROUTINE.
C
C      REL. HUMIDITY TREATMENT PER G. HANEL/1976 ADV. IN GEOPHYS.
C
C      FOR DIMENSION SIZES REFER TO THE BLOCK DATA SUBROUTINE. THERE
C      ALSO IS A ERROR ROUTINE (DIMER) THAT CKS ON YOUR DIMENSIONS.
C
C      SCATTERING FRACTIONS REQUIRE THAT NUNIT BE ASSIGNED AND A
C      SIMPLE CHANGE BE MADE IN SUBROUTINE AGXP3: FOR FURTHER INFO
C      REFER TO THAT SUBROUTINE.
C
C      THE FUNCTION ATAN2(SQRT(1.-C(I)**2),C(I)) IS EQUIVALENT
C      TO ARCOS(C(I)).
C
C      ***** ACKNOWLEDGEMENTS *****
C
C      THIS PROGRAM HAS BEEN CONSTRUCTED BY THE ATMOSPHERIC SCIENCES
C      LABORATORY AND NEW MEXICO STATE UNIVERSITY, DEPT OF PHYSICS. THE
C      FOLLOWING PEOPLE HAVE PARTICIPATED IN THIS UNDERTAKING:
C
C      DR. A.U. MILLER      NMSU
C      DR. R.C. SHIRKEY     ASL
C      DR. G.H. GOEDECKE    NMSU
C      MR. E.J. BURLEAW     NMSU
C
C      THE POINT OF CONTACT IS R.C. SHIRKEY, ASL.  PHONE (505) 678-5470
C      OR AV 258-5470.
C
C      *****
C      REAL KEXTT,KSCAT,KBAKT
C      COMMON /AGXM/ C(65),W(65),OLT(65),JDIMCK(3)
C      COMMON /PT1/ F(513),R(513),DR(8),RR(9),FF(514)
C      +,NRADI,PI,IDSTP,NKG,NHALV,NI
C      COMMON /PT2/ PC(65),OL(65),RMS(65),PSUM(65),PSUMT(65),P(65)
C      COMMON /IO/ IOIN,IOUT,NUNIT,IEO,NEOU
C      DIMENSION OUT(10,4),NTITLE(40)
C      ***** READ AND WRITE IDENTIFIER *****
C      READ (IOIN,88) (NTITLE(I),I=1,40)
C      FORMAT (40A2)
C      WRITE (IOUT,89) (NTITLE(I),I=1,40)
C      FORMAT (1H1,40A2//)
C      ***** READ INTEGER CONTROL PARAMETERS FOR THIS RUN *****
C      READ (IOIN,103) NWAVE,NINDX,IW,IDSTP,NRADI,IT,MORTE,IANG,IEO,NEOU
C      ERROR CHECKS
C      IF (IT.LE.0) IT=1
C      IF (IT.GT.JDIMCK(1)) CALL DIMER(1)
C      IF (JDIMCK(1).LT.65) WRITE(IOUT,129)
C      CHECK FOR CONFLICTING EOSAEL OPTIONS
C      IF ((IEO.EQ.2.OR.IEO.EQ.5).AND.NWAVE.EQ.1) IEO=1
C      IF ((IEO.EQ.4.OR.IEO.EQ.5).AND.NWAVE.EQ.1) IEO=3
C      IF (IEO.GT.1.AND.IT.GT.65) GO TO 20
C      EOSAEL OPTION
C      IF (IEO.EQ.1.OR.IEO.EQ.2.OR.IEO.EQ.5) IT=65
C      JDIMCK(3)=1+2*JDIMCK(2)
C      IF (IDSTP.GT.12) GO TO 1
C      IF (NWAVE.EQ.0) NWAVE=1
C      IF (IDSTP.EQ.12) IW=1
C      IF (NINDX.LT.1.OR.IDSTP.EQ.6.OR.IDSTP.EQ.12) NINDX=1
C      WRITE (IOUT,104) NWAVE,NINDX,IW,IDSTP,NRADI,IT,MORTE,IANG,IEO,NEOU
C      IF (IW.EQ.0) WRITE (IOUT,122)
C
AGX01380
AGX01390
AGX01400
AGX01410
AGX01420
AGX01430
AGX01440
AGX01450
AGX01460
AGX01470
AGX01480
AGX01490
AGX01500
AGX01510
AGX01520
AGX01530
AGX01540
AGX01550
AGX01560
AGX01570
AGX01580
AGX01590
AGX01600
AGX01610
AGX01620
AGX01630
AGX01640
AGX01650
AGX01660
AGX01670
AGX01680
AGX01690
AGX01700
AGX01710
AGX01720
AGX01730
AGX01740
AGX01750
AGX01760
AGX01770
AGX01780
AGX01790
AGX01800
AGX01810
AGX01820
AGX01830
AGX01840
AGX01850
AGX01860
AGX01870
AGX01880
AGX01890
AGX01900
AGX01910
AGX01920
AGX01930
AGX01940
AGX01950
AGX01960
AGX01970
AGX01980
AGX01990
AGX02000
AGX02010
AGX02020
AGX02030
AGX02040
AGX02050
AGX02060
AGX02070

```

```

C      INITIALIZE QUANTITIES USED IN SUMMATIONS
      DO 2 I=1,IT
      OLT(I)=0.
2     PSUMT(I)=0.E0
      WAVAVG=0.
      ALBDOT=0.
      KEXTT=0.E0
      KSCAT=0.E0
      KBAKT=0.E0
      CATTH=0.E0
      ITT=IT-1
      PI=3.1415926535898E+00
      IF (IEO.EQ.1.OR.IEO.EQ.2.OR.IEO.EQ.5) GO TO 3
      IF ((IANG.EQ.1).OR.(IANG.EQ.2)) GO TO 3
C      WHEN IANG=0 ROUTINE GUSET IS CALLED TO SET-UP THE ABSCISSAE AND
C      WEIGHTS USED FOR CALCULATING THE PHASE-FUNCTION AT -IT- POINTS
C      USED FOR NUMERICAL INTEGRATION VIA GAUSS-LEGENDRE QUADRATURE AND
C      THE PHASE FUNCTION EXPANSION COEFS, OL( ).
C      THE WEIGHTS ARE PLACED IN THE ARRAY W( ), AND THE COSINES OF THE
C      ANGLES ARE PLACED IN THE ARRAY C( ).
      CALL GUSET(IT)
      IF (ITT.LT.3) ITT=3
      GO TO 7
3     CALL ANGLE (PI,IANG,ITT)
C      SUBROUTINE ANGLE IS CALLED WHEN IANG=1 OR 2 TO SET UP THE
C      ANGLES AT WHICH PHASE FUNCTIONS WILL BE CALCULATED.  ANGLES
C      GO INTO ARRAY W( ) AND COSINES IN C( ).
      CONTINUE
      WRITE ANGLES FOR EOSAEL DATA FILE
      IF (IEO.LE.0) GO TO 21
      DO 22 I=1,ITT
22     C(I)=180.*ATAN2(SQRT(1.-C(I)**2),C(I))/PI
      ITP1=IT+1
      IF (ITP1.GT.JDIMCK(1)) ITP1=IT
      IF (ITT.LT.65.AND.JDIMCK(1).GT.65) C(ITP1)=999.99
      WRITE (NEOU,125) (C(I),I=1,ITP1)
      DO 23 I=1,ITT
23     C(I)=COS(C(I)*PI/180.)
21     CONTINUE
C      DETERMINE DETAILS OF AEROSOL SIZE-DISTRIBUTION VIA AGXP1
      CALL AGXP1(DENS,FSUM,VOL,JDIMCK)
      IF (IDSTP.EQ.6.OR.IDSTP.EQ.12) ELWC=DENS
C      DRYVOL IS THE AVERAGE VOLUME OF THE DRY AEROSOL PARTICLES IN
C      CUBIC MICROMETERS.
      DRYVOL=VOL
C      *** READ INPUT PARAMETERS ***
      READ (IOIN,105) WAVE,DWAVE,RELHUM,DENSH,TEMP,DELTA
      IF (NWAVE.EQ.1) DWAVE=0.E0
      WRITE (IOUT,106) WAVE,DWAVE,RELHUM,DENSH,TEMP,DELTA
      IF (NINDX.GT.1) WRITE (IOUT,107) NINDX
      IF ((DWAVE.LT.1E-04).AND.(NWAVE.GT.1)) WRITE (IOUT,108) NWAVE
      IF ((DWAVE.GE.1E-04).AND.(NWAVE.GT.1)) WRITE (IOUT,109) NWAVE
      IF (DENSH.LT.1E-04) WRITE (IOUT,110)
      ENWAY=FLOAT(NWAVE)
      IF (DWAVE.LT.1.E-4) GO TO 8
      WAVE=WAVE-DWAVE
8     DO 9 NWV=1,NWAVE
      IF (DWAVE.GT.1.E-4) GO TO 10
      IF (NWV.EQ.1) GO TO 11
      READ (IOIN,105) RELHUM,TEMP
      GO TO 11
10     WAVE=WAVE+DWAVE
11     VOL=DRYVOL
C      DETERMINE WHETHER THE USER SUPPLIED PARTICLE NUMBER DENSITY DENSH
C      SHOULD BE OVERRIDDEN BECAUSE THE CHOSEN IDSTP CASE HAS FIXED
C      PARAMETERS, AND/OR IF NUMBER DENSITIES ARE TO BE CALCULATED LATER
C      FROM THE AVG PARTICLE VOLUME, MASS DENSITY, AND MASS CONCENTRATION
      LLLL=0
      IF (IDSTP.EQ.6) GO TO 12
      IF (IDSTP.EQ.3.OR.IDSTP.GE.7) LLLL=1

```

```

AGX02080
AGX02090
AGX02100
AGX02110
AGX02120
AGX02130
AGX02140
AGX02150
AGX02160
AGX02170
AGX02180
AGX02190
AGX02200
AGX02210
AGX02220
AGX02230
AGX02240
AGX02250
AGX02260
AGX02270
AGX02280
AGX02290
AGX02300
AGX02310
AGX02320
AGX02330
AGX02340
AGX02350
AGX02360
AGX02370
AGX02380
AGX02390
AGX02400
AGX02410
AGX02420
AGX02430
AGX02440
AGX02450
AGX02460
AGX02470
AGX02480
AGX02490
AGX02500
AGX02510
AGX02520
AGX02530
AGX02540
AGX02550
AGX02560
AGX02570
AGX02580
AGX02590
AGX02600
AGX02610
AGX02620
AGX02630
AGX02640
AGX02650
AGX02660
AGX02670
AGX02680
AGX02690
AGX02700
AGX02710
AGX02720
AGX02730
AGX02740
AGX02750
AGX02760
AGX02770

```

```

IF (LLLL.EQ.1) GO TO 12
IF (DENSH.LE.1.E-4) GO TO 12
LLLL=1
DENS=DENSH
12 CONTINUE
RESTRICT RELATIVE HUMIDITY TO MAX OF 99 PERCENT.
IF (RELHUM.GE.99.E+00) RELHUM=99.0E+00
WRITE (IOUT,111) RELHUM,WAVE
IF (DENS.EQ.0.0) DENS=1.0E+00
GNU=1.0E+04/WAVE
IF (IDSTP.EQ.6.OR.IDSTP.EQ.12) DENS=ELWC
DENS IS USED AS AN ALIAS TO PASS ELWC TO ROUTINE AGXP2.
CALL AGXP2(RELHUM,CTSUM,CSSUM,CRSUM,VOL,TMASS,DENS,CATTN,TEMP,
1 DELTA,NINDX,IW,OLSTAR,OM2,LLLL,IT,WAVE,EM,CAY,EMM,MQRTE,PFNZRO)
LMAX=3*IFIX(2.E+0*PI*EMM*R(NRADI)/WAVE)
IF (LMAX.GT.IT) WRITE (IOUT,112) LMAX,IT
13 CALL AGXP3(CTSUM,CSSUM,CRSUM,GNU,DENS,NINDX,WAVE,EM,CAY,EMM,IT,0,
+IANG)
SUM QUANTITIES OVER INDEX NWV.
DO 14 IK=1,IT
OL(IK)=OL(IK)+OL(IK)
14 PSUM(IK)=PSUM(IK)+PSUM(IK)
ALBDOT BECOMES THE TOTAL SINGLE SCATTERING ALBEDO
KEXTT BECOMES THE TOTAL EXTINCTION COEF. (PER KILOMETER)
KSCAT BECOMES THE TOTAL SCATTERING COEF. (PER KM)
KBAKT BECOMES THE TOTAL BACK-SCATTERING COEF (PER KM)
ARRAY OUT(,) HOLDS SOME QUANTITIES FOR LATER PRINTOUTS
ALBDO=CSSUM/CTSUM
ALBDOT=ALBDOT+ALBDO
KEXTT=KEXTT+CTSUM
KSCAT=KSCAT+CSSUM
KBAKT=KBAKT+CRSUM
CATTN=CATTN+QATTN
WAVAVG=WAVAVG+WAVE
OUT(NWV,1)=WAVE
OUT(NWV,2)=RELHUM
OUT(NWV,3)=TMASS*1.E5
OUT(NWV,4)=CTSUM
IF ((NWAVE.GT.1).AND.(DWAVE.GE.1.E-04)) WRITE (IOUT,113) NWV
IF ((NWAVE.GT.1).AND.(DWAVE.LT.1.E-04)) WRITE (IOUT,114) NWV
EOSAEL OPTION: WRITE NBR OF ANGLES, WAVELENGTH, SINGLE SCATTERING
ALBEDO, EXTINCTION COEFFICIENT (TOTAL AND SCATTERING) FOR
INDIVIDUAL WAVELENGTHS.
IF (IEO.EQ.1.OR.IEO.EQ.3.OR.IEO.EQ.5) WRITE (NEOU,127)
+IT,WAVE,ALBDO,CTSUM,CSSUM
EOSAEL OPTION: WRITE PHASE FUNCTION FOR INDIVIDUAL WAVELENGTHS.
IF (IEO.EQ.1.OR.IEO.EQ.3.OR.IEO.EQ.5) WRITE (NEOU,128)
+ (PSUM(I),I=1,IT)
CONTINUE
END OF NWAVE LOOP
IF (NWAVE.LE.1) GO TO 19
DIVIDE BY NBR OF VALUES OF NWV TO GET AVERAGED RESULTS
DO 16 I=1,IT
OL(I)=OL(I)/ENWAV
PSUM(I)=PSUM(I)/ENWAV
16 CONTINUE
ALBDOT=ALBDOT/ENWAV
KEXTT=KEXTT/ENWAV
KSCAT=KSCAT/ENWAV
KBAKT=KBAKT/ENWAV
CATTN=CATTN/ENWAV
WAVAVG=WAVAVG/ENWAV
WRITE (IOUT,117) NWAVE
WRITE (IOUT,118)
DO 18 J=1,NWAVE
WRITE (IOUT,119) (OUT(J,JJ),JJ=1,4)
18 WRITE (IOUT,123) KEXTT,KSCAT,KBAKT,CATTN,ALBDOT
CALL AGXP3(CTSUM,CSSUM,CRSUM,GNU,DENS,NINDX,WAVE,EM,CAY,EMM,IT,1,
+IANG)
EOSAEL OPTION: WRITE NBR OF ANGLES, WAVELENGTH, SINGLE SCATTERING

```

```

C      ALBEDO, EXTINCTION COEFFICIENT (TOTAL AND SCATTERING) FOR      AGX03480
C      COMPOSITE VALUES      AGX03490
      IF (IEO.EQ.2.OR.IEO.EQ.4.OR.IEO.EQ.5) WRITE (NEOU,127)      AGX03500
      +      IT,WAVAVG,ALBDOT,KEXTT,KSCAT      AGX03510
C      EOSAEL OPTION: WRITE COMPOSITE PHASE FUNCTION.      AGX03520
      IF (IEO.EQ.2.OR.IEO.EQ.4.OR.IEO.EQ.5) WRITE (NEOU,128)      AGX03530
      +      (PSUM(I),I=1,IT)      AGX03540
      GO TO 19      AGX03550
      1 WRITE (IOUT,120) IDSTP      AGX03560
      19 WRITE (IOUT,126)      AGX03570
      STOP      AGX03580
20      WRITE (IOUT,124)      AGX03590
      STOP      AGX03600
      103 FORMAT (10I5)      AGX03610
      104 FORMAT(1H,78HINTEGER CONTROL PARAMETERS: NWAVE NINDX IW IDSTP NRA      AGX03620
      +      OI IT MORTE IANG IEO NEOU/,1X,29X,2(I2,4X),11,3X,12,      AGX03630
      +      3X,13,1X,13,1X,15,1X,15,2(2X,12))      AGX03640
      105 FORMAT (6E12.6)      AGX03650
      106 FORMAT(1H,/,17H INPUT PARAMETERS/,1X,6X,9HWAVE = ,E12.6,8H MICR      AGX03660
      +      ONS/,1X,6X,9HDWAVE = ,E12.6,8H MICRONS/,1X,6X,9HRELHUM = ,E12.6,      AGX03670
      +      3H PERCENT/,1X,6X,9HDENSH = ,E12.6,13H PARTICLES/CC/,1X,6X,      AGX03680
      +      9HTEMP = ,E12.6,6H DEG C/,1X,6X,19HDELTA (CONVERGENCE ,      AGX03690
      +      13HCRITERION) = ,E12.6)      AGX03700
      107 FORMAT (/,1H,29HLOOPING OPTION IN EFFECT FOR ,12,      AGX03710
      +      19H AEROSOL COMPONENTS)      AGX03720
      108 FORMAT (/,1H,39HRELATIVE HUMIDITY OPTION IN EFFECT FOR ,12,      AGX03730
      +      7H VALUES)      AGX03740
      109 FORMAT (/,1H,40HWAVELENGTH LOOPING OPTION IN EFFECT FOR ,12,      AGX03750
      +      12H WAVELENGTHS)      AGX03760
      110 FORMAT(1H,52H*** PARTICLE NUMBER DENSITY WILL BE CALCULATED FROM      AGX03770
      +      41H MASS DENSITY AND MASS CONCENTRATION ***)      AGX03780
      111 FORMAT (1H1,/,1X,33HRELATIVE HUMIDITY FOR THIS RUN = ,F6.2,      AGX03790
      +      25H PERCENT, WAVELENGTH = ,F10.3,8H MICRONS,/)      AGX03800
      112 FORMAT (/,49H *** WARNING *** OPTIMAL PF EXPANSION ORDER OF ,13,      AGX03810
      +      22H EXCEEDS INPUT IT = ,13,24H. PF VALUES SHOULD BE ,      AGX03820
      +      15HUSED CAUTIOUSLY/)      AGX03830
      113 FORMAT (1H,/,1X,40(1H*),3X,31HEND OF WAVELENGTH CYCLE NUMBER ,13,      AGX03840
      +      3X,40(1H*))      AGX03850
      114 FORMAT (1H,/,1X,40(1H*),3X,38HEND OF RELATIVE HUMIDITY CYCLE NUMB      AGX03860
      +      ER ,13,3X,40(1H*))      AGX03870
      117 FORMAT(1H1,/,47H SUMMARY OF RESULTS FOR THIS RUN AVERAGED OVER ,      AGX03880
      +      12,30H WAVELENGTH(S) ARE AS FOLLOWS:/)      AGX03890
      118 FORMAT(1H,4X,48HWAVELENGTH REL.HUMIDITY AEROSOL MASS K(,AGX03900
      +      11HEXTINCTION),/1X,29H (MICROMETERS) (PERCENT)      AGX03910
      +      6H (GM,25H/(SQ.CM-KM) (PER KM)/)      AGX03920
      119 FORMAT (2F15.6,1P2E16.5)      AGX03930
      120 FORMAT (/,13H *** IDSTP = ,15,35H IS ILLEGAL. EXECUTION TERMINATED      AGX03940
      +      2H*/)      AGX03950
      122 FORMAT (/,1X,23H*** WATER ONLY CASE ***/)      AGX03960
      123 FORMAT (/,20H EXTINCTION COEF. = ,5X,1PE13.7,9H (PER KM),/,      AGX03970
      120H SCATTERING COEF. = ,8X,1PE13.7,9H (PER KM),/,      AGX03980
      225H BACK-SCATTERING COEF. = ,3X,1PE13.7,9H (PER KM),/,      AGX03990
      321H ATTENUATION COEF. = ,7X,1PE13.7,13H SQ-METERS/MG,/,      AGX04000
      428H SINGLE SCATTERING ALBEDO = ,1PE13.7,/)      AGX04010
      124 FORMAT(1H,58H*** MORE THAN 65 ANGLES FOR EOSAEL OPTION - PGM TERM      AGX04020
      +      INATED)      AGX04030
      125 FORMAT (11(F6.2,1X))      AGX04040
      126 FORMAT(1H1)      AGX04050
      127 FORMAT(12,1X,2H00,1X,F5.2,1X,F8.6,1X,2(E12.6,1X))      AGX04060
      128 FORMAT(6(E12.6,1X))      AGX04070
      129 FORMAT(1H,23H*** AGAUS WARNING ****,/,1X,      AGX04080
      +      37HTHE ARRAY W IS ASSIGNED 65 VALUES IN,/,1X,      AGX04090
      +      43HBLOCK DATA WHICH IS LARGER THAN ARRAY SIZE,/,1X,      AGX04100
      +      47HYOU MAY BE CLOBBING INSTRUCTIONS AND/OR DATA.//)      AGX04110
      END      AGX04120

```


AD-A114 417 ARMY ELECTRONICS RESEARCH AND DEVELOPMENT COMMAND WS--ETC F/G 4/1
PROGRAM LISTINGS FOR EOSAEL 80-8 AND ANCILLARY CODES AGAUS AND --ETC(11)
FEB 82 R G STEINHOFF
UNCLASSIFIED ERADCOM/ASL-TR-0107-V2-SU NL

5 6

21

11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674 675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760 761 762 763 764 765 766 767 768 769 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816 817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 860 861 862 863 864 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 900 901 902 903 904 905 906 907 908 909 910 911 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977 978 979 980 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998 999 1000 1001 1002 1003 1004 1005 1006 1007 1008 1009 1010 1011 1012 1013 1014 1015 1016 1017 1018 1019 1020 1021 1022 1023 1024 1025 1026 1027 1028 1029 1030 1031 1032 1033 1034 1035 1036 1037 1038 1039 1040 1041 1042 1043 1044 1045 1046 1047 1048 1049 1050 1051 1052 1053 1054 1055 1056 1057 1058 1059 1060 1061 1062 1063 1064 1065 1066 1067 1068 1069 1070 1071 1072 1073 1074 1075 1076 1077 1078 1079 1080 1081 1082 1083 1084 1085 1086 1087 1088 1089 1090 1091 1092 1093 1094 1095 1096 1097 1098 1099 1100 1101 1102 1103 1104 1105 1106 1107 1108 1109 1110 1111 1112 1113 1114 1115 1116 1117 1118 1119 1120 1121 1122 1123 1124 1125 1126 1127 1128 1129 1130 1131 1132 1133 1134 1135 1136 1137 1138 1139 1140 1141 1142 1143 1144 1145 1146 1147 1148 1149 1150 1151 1152 1153 1154 1155 1156 1157 1158 1159 1160 1161 1162 1163 1164 1165 1166 1167 1168 1169 1170 1171 1172 1173 1174 1175 1176 1177 1178 1179 1180 1181 1182 1183 1184 1185 1186 1187 1188 1189 1190 1191 1192 1193 1194 1195 1196 1197 1198 1199 1200 1201 1202 1203 1204 1205 1206 1207 1208 1209 1210 1211 1212 1213 1214 1215 1216 1217 1218 1219 1220 1221 1222 1223 1224 1225 1226 1227 1228 1229 1230 1231 1232 1233 1234 1235 1236 1237 1238 1239 1240 1241 1242 1243 1244 1245 1246 1247 1248 1249 1250 1251 1252 1253 1254 1255 1256 1257 1258 1259 1260 1261 1262 1263 1264 1265 1266 1267 1268 1269 1270 1271 1272 1273 1274 1275 1276 1277 1278 1279 1280 1281 1282 1283 1284 1285 1286 1287 1288 1289 1290 1291 1292 1293 1294 1295 1296 1297 1298 1299 1300 1301 1302 1303 1304 1305 1306 1307 1308 1309 1310 1311 1312 1313 1314 1315 1316 1317 1318 1319 1320 1321 1322 1323 1324 1325 1326 1327 1328 1329 1330 1331 1332 1333 1334 1335 1336 1337 1338 1339 1340 1341 1342 1343 1344 1345 1346 1347 1348 1349 1350 1351 1352 1353 1354 1355 1356 1357 1358 1359 1360 1361 1362 1363 1364 1365 1366 1367 1368 1369 1370 1371 1372 1373 1374 1375 1376 1377 1378 1379 1380 1381 1382 1383 1384 1385 1386 1387 1388 1389 1390 1391 1392 1393 1394 1395 1396 1397 1398 1399 1400 1401 1402 1403 1404 1405 1406 1407 1408 1409 1410 1411 1412 1413 1414 1415 1416 1417 1418 1419 1420 1421 1422 1423 1424 1425 1426 1427 1428 1429 1430 1431 1432 1433 1434 1435 1436 1437 1438 1439 1440 1441 1442 1443 1444 1445 1446 1447 1448 1449 1450 1451 1452 1453 1454 1455 1456 1457 1458 1459 1460 1461 1462 1463 1464 1465 1466 1467 1468 1469 1470 1471 1472 1473 1474 1475 1476 1477 1478 1479 1480 1481 1482 1483 1484 1485 1486 1487 1488 1489 1490 1491 1492 1493 1494 1495 1496 1497 1498 1499 1500 1501 1502 1503 1504 1505 1506 1507 1508 1509 1510 1511 1512 1513 1514 1515 1516 1517 1518 1519 1520 1521 1522 1523 1524 1525 1526 1527 1528 1529 1530 1531 1532 1533 1534 1535 1536 1537 1538 1539 1540 1541 1542 1543 1544 1545 1546 1547 1548 1549 1550 1551 1552 1553 1554 1555 1556 1557 1558 1559 1560 1561 1562 1563 1564 1565 1566 1567 1568 1569 1570 1571 1572 1573 1574 1575 1576 1577 1578 1579 1580 1581 1582 1583 1584 1585 1586 1587 1588 1589 1590 1591 1592 1593 1594 1595 1596 1597 1598 1599 1600 1601 1602 1603 1604 1605 1606 1607 1608 1609 1610 1611 1612 1613 1614 1615 1616 1617 1618 1619 1620 1621 1622 1623 1624 1625 1626 1627 1628 1629 1630 1631 1632 1633 1634 1635 1636 1637 1638 1639 1640 1641 1642 1643 1644 1645 1646 1647 1648 1649 1650 1651 1652 1653 1654 1655 1656 1657 1658 1659 1660 1661 1662 1663 1664 1665 1666 1667 1668 1669 1670 1671 1672 1673 1674 1675 1676 1677 1678 1679 1680 1681 1682 1683 1684 1685 1686 1687 1688 1689 1690 1691 1692 1693 1694 1695 1696 1697 1698 1699 1700 1701 1702 1703 1704 1705 1706 1707 1708 1709 1710 1711 1712 1713 1714 1715 1716 1717 1718 1719 1720 1721 1722 1723 1724 1725 1726 1727 1728 1729 1730 1731 1732 1733 1734 1735 1736 1737 1738 1739 1740 1741 1742 1743 1744 1745 1746 1747 1748 1749 1750 1751 1752 1753 1754 1755 1756 1757 1758 1759 1760 1761 1762 1763 1764 1765 1766 1767 1768 1769 1770 1771 1772 1773 1774 1775 1776 1777 1778 1779 1780 1781 1782 1783 1784 1785 1786 1787 1788 1789 1790 1791 1792 1793 1794 1795 1796 1797 1798 1799 1800 1801 1802 1803 1804 1805 1806 1807 1808 1809 1810 1811 1812 1813 1814 1815 1816 1817 1818 1819 1820 1821 1822 1823 1824 1825 1826 1827 1828 1829 1830 1831 1832 1833 1834 1835 1836 1837 1838 1839 1840 1841 1842 1843 1844 1845 1846 1847 1848 1849 1850 1851 1852 1853 1854 1855 1856 1857 1858 1859 1860 1861 1862 1863 1864 1865 1866 1867 1868 1869 1870 1871 1872 1873 1874 1875 1876 1877 1878 1879 1880 1881 1882 1883 1884 1885 1886 1887 1888 1889 1890 1891 1892 1893 1894 1895 1896 1897 1898 1899 1900 1901 1902 1903 1904 1905 1906 1907 1908 1909 1910 1911 1912 1913 1914 1915 1916 1917 1918 1919 1920 1921 1922 1923 1924 1925 1926 1927 1928 1929 1930 1931 1932 1933 1934 1935 1936 1937 1938 1939 1940 1941 1942 1943 1944 1945 1946 1947 1948 1949 1950 1951 1952 1953 1954 1955 1956 1957 1958 1959 1960 1961 1962 1963 1964 1965 1966 1967 1968 1969 1970 1971 1972 1973 1974 1975 1976 1977 1978 1979 1980 1981 1982 1983 1984 1985 1986 1987 1988 1989 1990 1991 1992 1993 1994 1995 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020 2021 2022 2023 2024 2025 2026 2027 2028 2029 2030 2031 2032 2033 2034 2035 2036 2037 2038 2039 2040 2041 2042 2043 2044 2045 2046 2047 2048 2049 2050 2051 2052 2053 2054 2055 2056 2057 2058 2059 2060 2061 2062 2063 2064 2065 2066 2067 2068 2069 2070 2071 2072 2073 2074 2075 2076 2077 2078 2079 2080 2081 2082 2083 2084 2085 2086 2087 2088 2089 2090 2091 2092 2093 2094 2095 2096 2097 2098 2099 2100 2101 2102 2103 2104 2105 2106 2107 2108 2109 2110 2111 2112 2113 2114 2115 2116 2117 2118 2119 2120 2121 2122 2123 2124 2125 2126 2127 2128 2129 2130 2131 2132 2133 2134 2135 2136 2137 2138 2139 2140 2141 2142 2143 2144 2145 2146 2147 2148 2149 2150 2151 2152 2153 2154 2155 2156 2157 2158 2159 2160 2161 2162 2163 2164 2165 2166 2167 2168 2169 2170 2171 2172 2173 2174 2175 2176 2177 2178 2179 2180 2181 2182 2183 2184 2185 2186 2187 2188 2189 2190 2191 2192 2193 2194 2195 2196 2197 2198 2199 2200 2201 2202 2203 2204 2205 2206 2207 2208 2209 2210 2211 2212 2213 2214 2215 2216 2217 2218 2219 2220 2221 2222 2223 2224 2225 2226 2227 2228 2229 2230 2231 2232 2233 2234 2235 2236 2237 2238 2239 2240 2241 2242 2243 2244 2245 2246 2247 2248 2249 2250 2251 2252 2253 2254 2255 2256 2257 2258 2259 2260 2261 2262 2263 2264 2265 2266 2267 2268 2269 2270 2271 2272 2273 2274 2275 2276 2277 2278 2279 2280 2281 2282 2283 2284 2285 2286 2287 2288 2289 2290 2291 2292 2293 2294 2295 2296 2297 2298 2299 2300 2301 2302 2303 2304 2305 2306 2307 2308 2309 2310 2311 2312 2313 2314 2315 2316 2317 2318 2319 2320 2321 2322 2323 2324 2325 2326 2327 2328 2329 2330 2331 2332 2333 2334 2335 2336 2337 2338 2339 2340 2341 2342 2343 2344 2345 2346 2347 2348 2349 2350 2351 2352 2353 2354 2355 2356 2357 2358 2359 2360 2361 2362 2363 2364 2365 2366 2367 2368 2369 2370 2371 2372 2373 2374 2375 2376 2377 2378 2379 2380 2381 2382 2383 2384 2385 2386 2387 2388 2389 2390 2391 2392 2393 2394 2395 2396 2397 2398 2399 2400 2401 2402 2403 2404 2405 2406 2407 2408 2409 2410 2411 2412 2413 2414 2415 2416 2417 2418 2419 2420 2421 2422 2423 2424 2425 2426 2427 2428 2429 2430 2431 2432 2433 2434 2435 2436 2437 2438 2439 2440 2441 2442 2443 2444 2445 2446 2447 2448 2449 2450 2451 2452 2453 2454 2455 2456 2457 2458 2459 2460 2461 2462 2463 2464 2465 2466 2467 2468 2469 2470 2471 2472 2473 2474 2475 2476 2477 2478 2479 2480 2481 2482 2483 2484 2485 2486 2487 2488 2489 2490 2491 2492 2493 2494 2495 2496 2497 2498 2499 2500 2501 2502 2503 2504 2505 2506 2507 2508 2509 2510 2511 2512 2513 2514 2515 2516 2517 2518 2519 2520 2521 2522 2523 2524 2525 2526 2527 2528 2529 2530 2531 2532 2533 2534 2535 2536 2537 2538 2539 2540 2541 2542 2543 2544 2545 2546 2547 2548 2549 2550 2551 2552 2553 2554 2555 2556 2557 2558 2559 2560 2561 2562 2563 2564 2565 2566 2567 2568 2569 2570 2571 2572 2573 2574 2575 2576 2577 2578 2579 2580 2581 2582 2583 2584 2585 2586 2587 2588 2589 2590 2591 2592 2593 2594 2595 2596 2597 2598 2599 2600 2601 2602 2603 2604 2605 2606 2607 2608 2609 2610 2611 2612 2613 2614 2615 2616 2617 2618 2619 2620 2621 2622 2623 2624 2625 2626 2627 2628 2629 2630 2631 2632 2633 2634 2635 2636 2637 2638 2639 2640 2641 2642 2643 2644 2645 2646 2647 2648 2649 2650 2651 2652 2653 2654 2655 2656 2657

```

SUBROUTINE AGXP1 (DENS,FSUM,VOL,JDIMCK)
COMMON /PT1/ F(513),R(513),DR(8),RR(9),FF(514)
+ ,NRADI,PI,IDSTP,NKG,NHALV,NI
COMMON /IO/ IOIN,IOUT,NUNIT,IEO,NEOU
DIMENSION JDIMCK(3)
EXTERNAL GAMMA
WRITE (IOUT,2)
C CHOOSE AND SET UP PARTICLE SIZE DISTRIBUTION
IF (IDSTP.NE.0) GO TO (12,15,16,17,19,19,17,20,21,22,24,26),IDSTP
C** TYPE 0: ARBITRARY USER-SUPPLIED DISTRIBUTION. NRADI VALUES OF
C R(J) AND FF(J) MUST BE GIVEN, ONE PER CARD, AND READ IN ORDER
C FROM SMALLEST RADIUS, RLO TO THE LARGEST.
C NRADI MUST BE LESS THAN OR EQUAL TO 1+2**JDIMCK(2)
WRITE (IOUT,7)
IF (NRADI.GT.JDIMCK(3)) CALL DIMER(2)
DO 9 J=1,NRADI
9 READ (IOIN,1) R(J),FF(J)
RLO=R(1)
DO 10 J=1,NRADI,5
JK=J+4
IF (JK.GT.NRADI) JK=NRADI
10 WRITE (IOUT,11) (R(K),FF(K),K=J,JK)
WRITE (IOUT,11)
FF(NRADI+1)=FF(NRADI)
RR(1)=RLO
RR(2)=R(NRADI)
MIN=0
GO TO 28
C** TYPE 1: LOG-NORMAL DISTRIBUTION
12 READ (IOIN,1) RBAR,SIGMA,RLO,RHI
C SIGMA IS STANDARD DEVIATION, NOT LN(SIGMA)
SIGIN=SIGMA
SIGMA=ALOG(SIGMA)
A=ABS(1.E0/(2.5066283E0*SIGMA))
IF ((RHI-RLO).LE.1.E-4) GO TO 13
RR(1)=RLO
RR(3)=RHI
GO TO 14
13 RR(1)=RBAR*EXP(-4.E0*SIGMA)
RR(3)=RBAR*EXP(4.E0*SIGMA)
14 RR(2)=RBAR
MIN=1
WRITE (IOUT,3) RBAR,SIGIN,RLO,RHI
AVOL=4.18879E0*(RBAR**3.E0)*EXP(4.5E0*SIGMA*SIGMA)
C HERE AND ELSEWHERE, AVOL IS THE VOLUME OBTAINED VIA
C ANALYTICAL INTEGRATION OVER THE LIMITS RLO=0 TO RHI=
C INFINITY: THAT CAN ONLY BE DONE FOR A FEW IDSTP CASES.
GO TO 28
C** TYPE 2: DOUBLE EXPONENTIAL F(R)=CUE*A*EXP(-A*R)+(1-CUE)*B*EXP(-B*R).
C RESTRICTIONS: RHI.GT.RLO, B.GT.A,GE.0, 0.LE.CUE.LE.1.0.
15 READ (IOIN,1) RLO,RHI,CUE,A,B
WRITE (IOUT,4) RLO,RHI,CUE,A,B
RR(1)=RLO
RR(3)=RHI
RR(2)=0.5E0*(RLO+RHI)
MIN=1
GO TO 28
C** TYPE 3: DEIRMENDJIAN MODEL C. F(R) = 1.0, RLO.LE.R.LE.4*DELRD,
C F(R)=(4*DELRD/R)**4, R.GE.(4*DELRD)
C NRADI IS READ IN EARLIER IN THE MAIN PROGRAM.
16 DENS=1.378E+04
DELRD=0.02E0
RLO=0.02E0
RHI=RLO+DELRD* FLOAT(NRADI-1)
RR(1)=RLO
RR(3)=RHI
MIN=1
RR(2)=RLO+4.E0*DELRD
GO TO 28
C** TYPE 4 AND TYPE 7: POWER LAW. F(R) = CUE*R**-A

```

```

AGA00010
AGA00020
AGA00030
AGA00040
AGA00050
AGA00060
AGA00070
AGA00080
AGA00090
AGA00140
AGA00160
AGA00170
AGA00190
AGA00200
AGA00210
AGA00220
AGA00230
AGA00240
AGA00250
AGA00260
AGA00270
AGA00280
AGA00290
AGA00300
AGA00310
AGA00320
AGA00330
AGA00340
AGA00350
AGA00360
AGA00370
AGA00380
AGA00390
AGA00400
AGA00410
AGA00420
AGA00430
AGA00440
AGA00450
AGA00460
AGA00470
AGA00480
AGA00490
AGA00500
AGA00510
AGA00520
AGA00530
AGA00540
AGA00550
AGA00560
AGA00570
AGA00580
AGA00590
AGA00600
AGA00610
AGA00620
AGA00630
AGA00640
AGA00650
AGA00660
AGA00670
AGA00680
AGA00690
AGA00700
AGA00710

```

```

C      RLO,LE,R,LF,RHI; VIS=VISIBILITY IN KILOMETERS.
17  IF (IDSTP.EQ.4) GO TO 18
C**   TYPE 7 PRESCRIBED PARAMETER.
      READ (IOIN,1) VIS
      RLO=0.1E0
      RHI=15.E0
      CUE=30.E0
      A=4.E0
      DENS=11.E0**((5.E0-ALOG(0.1*VIS))
C**   TYPE 4 PRESCRIBED PARAMETERS.
18  IF (IDSTP.EQ.4) READ (IOIN,1) RLO,RHI,CUE,A
      WRITE (IOUT,5) RLO,RHI,CUE,A,VIS
      RR(1)=RLO
      RR(3)=RHI
      RR(2)=(0.5E0*(RLO**(-A)+RHI**(-A)))*(-1.E0/A)
      MIN=1
      GO TO 28
C**   TYPE 5: MODIFIED GAMMA/GENERALIZED KHIRGIAN-MAZIN
      F(R) = (R**ALF)*EXP(-ALF*((R/RC)**GAM)/GAM)
C      RLO,LE,R,LF,RHI.
C**   TYPE 6: SPECIAL CASE FOR WATER FOGS OR CLOUDS,
      IN WHICH CASE ELWC IS LIQUID WATER CONTENT
      IN GRAMS PER CUBIC CENTIMETER:
      ELWC IS IGNORED IF IDSTP = 5.
C      19 READ (IOIN,1) RLO,RHI,RC,ALF,GAM,ELWC
      IF (IDSTP.EQ.6) DENS=ELWC
      WRITE (IOUT,6) RLO,RHI,RC,ALF,GAM
      RR(1)=RLO
      RR(2)=RC
      RR(3)=RHI
      MIN=1
      B=ALF/(GAM*RC**GAM)
      AVOL=4.1888*B**(-3./GAM)*GAMMA((ALF+4.)/GAM)/GAMMA((ALF+1.)/GAM)
      GO TO 28
C**   TYPES 8,9,10: BIMODAL LOG-NORMAL DISTRIBUTIONS.
C      METHOD BELOW VALID FOR RBARC*EXP(-SGA).GT.RBARA*EXP(SGA)
C**   TYPE 8: CONTINENTAL BIMODAL.
20  FOA=4.E03
      FOC=2.1E0
      SGA=0.74E0
      SGC=0.81E0
      RBARA=0.03E0
      RBARC=0.4E0
      GO TO 23
C**   TYPE 9: MARITIME BIMODAL.
21  FOA=4.E02
      FOC=3.8E0
      SGA=0.68E0
      SGC=0.74E0
      RBARA=0.05E0
      RBARC=0.65E0
      GO TO 23
C**   TYPE 10: URBAN BIMODAL.
22  FOA=2.E04
      FOC=0.6E0
      SGA=0.63E0
      SGC=0.77E0
      RBARA=0.04E0
      RBARC=0.63E0
C      CALCULATE RADII FOR TYPES 8,9,10.
C      23 RR(1)=RBARA*EXP(-4.E0*ABS(SGA))
      RR(2)=RBARA
      RR(3)=RBARA*EXP(4.E0*ABS(SGA))
      RR(4)=RBARC*EXP(-4.E0*ABS(SGC))
      RR(5)=RBARC*EXP(4.E0*ABS(SGC))
      MIN=2
      DO 60 J=1,4
      DO 60 I=1,4
      IF (RR(I+1).GT.RR(I)) GO TO 60
      HH=RR(I)

```

```

AGA00720
AGA00730
AGA00740
AGA00750
AGA00760
AGA00770
AGA00780
AGA00790
AGA00800
AGA00810
AGA00820
AGA00830
AGA00840
AGA00850
AGA00860
AGA00870
AGA00880
AGA00890
AGA00900
AGA00910
AGA00920
AGA00930
AGA00940
AGA00950
AGA00960
AGA00970
AGA00980
AGA00990
AGA01000
AGA01010
AGA01020
AGA01030
AGA01040
AGA01050
AGA01060
AGA01070
AGA01080
AGA01090
AGA01100
AGA01110
AGA01120
AGA01130
AGA01140
AGA01150
AGA01160
AGA01170
AGA01180
AGA01190
AGA01200
AGA01210
AGA01220
AGA01230
AGA01240
AGA01250
AGA01260
AGA01270
AGA01280
AGA01290
AGA01300
AGA01310
AGA01320
AGA01330
AGA01340
AGA01350
AGA01360
AGA01370
AGA01380
AGA01390
AGA01400
AGA01410

```

```

RR(I)=RR(I+1)
RR(I+1)=HH
60 CONTINUE
GO TO 28
C** TYPE 11:
C USER SUPPLIED BIMODAL CASE: FOA AND FOC ARE THE NUMBER DENSITIES
C FOR THE ACCUMULATION (SMALLER RBAR) AND COARSE MODES,
C RESPECTIVELY, IN PARTICLES PER CUBIC CENTIMETER,
C SGA IS STD.DEVIATION FOR MODE A ** NOT LN(SIGMA) ***
C SGC IS STD.DEVIATION FOR MODE C ** NOT LN(SIGMA) ***
C *** NOTE, HOWEVER, THAT SGA AND SGC ARE THE LOGS OF THE
C STANDARD DEVIATIONS IN THE PRE-CODED CASES TYPE 8-10.
24 READ (IOIN,1) FOA,RBARA,SGA,FOC,RBARC,SGC
WRITE (IOUT,25) FOA,RBARA,SGA,FOC,RBARC,SGC
SGA=ABS(ALOG(SGA))
SGC=ABS(ALOG(SGC))
GO TO 23
C** TYPE 12: MARSHALL-PALMER RAIN MODEL,
C C.F. MASON, PHYSICS OF CLOUDS, CH. ON RADAR METEOROLOGY.
C INPUT PARAMETER RAIN IS RAIN RATE IN MILLIMETERS/HOUR:
C ** EMA,CAYA, AND RHOA ARE REQUIRED FOR THIS DISTRIBUTION.
26 READ (IOIN,1) RAIN
ENZERO=0.08E0
CAPL=41.E0*RAIN**(-0.21E0)
DENS=ENZERO/CAPL
AVOL=PI*(CAPL**(-3.E0))*1.E12
C CONVERT UNITS FROM CM-4 TO (CM-3)*(MICROMETERS**(-1)):
C THE FACTOR OF 2 CONVERTS THE M-P FORMULA FROM DIAMETER-DATA TO
C RADIUS BASED FORM.
ENZERO=2.E-4*ENZERO
CAPL=2.E-4*CAPL
MIN=0
RR(1)=1.E-4
RR(2)=2500.E0
WRITE (IOUT,27) RAIN,DENS
C THE NEXT BLOCK IS COMMON TO ALL DISTRIBUTIONS.
C IT SETS THE NMAX VALUES OF RADIUS, R(KK).
28 MAX=JDIMCK(2)
NHALV=MAX-MIN
NMAX=1+2**MAX
NI=2**MIN
IF (NMAX.GT.JDIMCK(3).OR.NI.GT.JDIMCK(3)) CALL DIMER(3)
NLAST=NI+1
NKG=2**NHALV
ENKG=FLOAT(NKG)
IF (IDSTP.EQ.0) GO TO 30
DO 29 I=1,NI
DR(I)=RR(I+1)-RR(I)
DO 29 K=1,NKG
KK=(I-1)*NKG+K
29 R(KK)=RR(I)+(FLOAT(K-1)*DR(I))/ENKG
R(NMAX)=RR(NLAST)
C BRANCH AGAIN CALCULATE THE DIFFERENT F(R) ON THE NMAX POINTS R(K)
GO TO (31,33,35,38,41,41,39,43,43,43,43,46),IDSTP
C** TYPE 0: ARBITRARY
C INTERPOLATE TO EQUAL INCREMENTS OVER RADII
30 DELR=(R(NRADI)-RLO)/ENKG
F(1)=FF(1)
NMAXM1=NMAX-1
DO 64 KK=1,NMAXM1
RADIUS=RLO+DELR*FLOAT(KK)
DO 62 J=1,NRADI
K=J
IF (R(J).GE.RADIUS) GO TO 61
62 CONTINUE
61 CONTINUE
F(KK+1)=(RADIUS-R(K-1))*(FF(K)-FF(K-1))/
(R(K)-R(K-1))+FF(K-1)
64 CONTINUE
DO 65 I=1,NI

```

```

AGA01420
AGA01430
AGA01440
AGA01450
AGA01460
AGA01470
AGA01480
AGA01490
AGA01500
AGA01510
AGA01520
AGA01530
AGA01540
AGA01550
AGA01560
AGA01570
AGA01580
AGA01590
AGA01600
AGA01610
AGA01620
AGA01630
AGA01640
AGA01650
AGA01660
AGA01670
AGA01680
AGA01690
AGA01700
AGA01710
AGA01720
AGA01730
AGA01740
AGA01750
AGA01760
AGA01770
AGA01780
AGA01790
AGA01800
AGA01810
AGA01820
AGA01830
AGA01840
AGA01850
AGA01860
PGA01870
AGA01880
AGA01890
AGA01900
AGA01910
AGA01920
AGA01930
AGA01940
AGA01950
AGA01960
AGA01870

```

```

DR(I)=RR(I+1)-RR(I)
DO 65 K=1,NKG
KK=(I-1)*NKG+K
65 R(KK)=RR(I)+(FLOAT(K-1))*DR(I)/ENKG
R(NMAX)=RR(NLAST)
GO TO 48
C** TYPE 1: LOG NORMAL
31 DEN=2.E0*SIGMA*SIGMA
DO 32 KK=1,NMAX
GNUM=ALOG(R(KK)/RBAR)
32 F(KK)=EXP(-GNUM*GNUM/DEN)*A/R(KK)
GO TO 48
C** TYPE 2: DOUBLE EXPONENTIAL
33 DO 34 KK=1,NMAX
FKK=(1.E0-CUE)*B*EXP(-B*R(KK))
34 F(KK)=FKK+CUE*A*EXP(-A*R(KK))
GO TO 48
C** TYPE 3: DEIRMENDJIAN MODEL C.
35 DO 36 KK=1,NMAX
F(KK)=1.E0
NKG1=NKG+1
DO 37 KK=NKG1,NMAX
37 F(KK)=(RR(2)/R(KK))**4.E0
GO TO 48
C** TYPES 4 AND 7: POWER LAW
38 GO TO 39
39 DO 40 KK=1,NMAX
40 F(KK)=CUE*R(KK)**(-A)
GO TO 48
C** TYPE 5 AND TYPE 6: MODIFIED GAMMA
41 DO 42 KK=1,NMAX
42 F(KK)=(EXP(-B*R(KK))*GAM)**R(KK)*ALF
GO TO 48
C** TYPES 8,9,10,11: BIMODAL LOG-NORMAL DISTRIBUTIONS
43 DENA=2.E0*SGA*SGA
DENC=2.E0*SGC*SGC
FAA=FOA/SGA
FCC=FOC/SGC
DO 44 KK=1,NMAX
GNUMA=ALOG(R(KK)/RBARA)
GNUMC=ALOG(R(KK)/RBARC)
FA=FAA*EXP(-GNUMA*GNUMA/DENA)
FC=FCC*EXP(-GNUMC*GNUMC/DENC)
44 F(KK)=(FA+FC)/R(KK)
DENS=FOA+FOC
WRITE(IOUT,45) DENS
VOLA=4.18879E0*(RBARA**3.E0)*EXP(4.5E0*SGA*SGA)*FOA
VOLC=4.18879E0*(RBARC**3.E0)*EXP(4.5E0*SGC*SGC)*FOC
AVOL=(VOLA+VOLC)/DENS
GO TO 48
C** TYPE 12: MARSHALL-PALMER RAIN MODEL
46 DO 47 KK=1,NMAX
47 F(KK)=ENZERO*EXP(-CAPL*R(KK))
C CALCULATE NORMALIZED F(KK) AND SOME DRY VOLUMES USING ALL NMAX
C VALUES OF RADII.
C (VOL=AVERAGE PARTICLE VOLUME IN A DISTRIBUTION). THE
C NORMALIZATION AND FURTHER VOLUMES ARE RECALCULATED LATER
C BY THE HALVING INTEGRATION METHOD.
48 FSUM=0.E0
IF(F(1).LT.0.E0)F(1)=0.0E0
DO 49 J=2,NMAX
IF(F(J).LT.0.E0)F(J)=0.E0
49 FSUM=FSUM+0.5E0*(F(J)+F(J-1))*(R(J)-R(J-1))
DO 50 J=1,NMAX
F(J)=F(J)/FSUM
50 WRITE(IOUT,8) FSUM
NRADI=NMAX
IF(IDSTP.EQ.1.OR.IDSTP.EQ.5.OR.IDSTP.GE.8) WRITE(IOUT,51) AVOL
VOL=0.E0
DO 52 J=2,NMAX

```

```

AGA01880
AGA01890
AGA01900
AGA01910
AGA01920
AGA02010
AGA02020
AGA02030
AGA02040
AGA02050
AGA02060
AGA02070
AGA02080
AGA02090
AGA02100
AGA02110
AGA02120
AGA02130
AGA02140
AGA02150
AGA02160
AGA02170
AGA02180
AGA02190
AGA02200
AGA02210
AGA02220
AGA02230
AGA02240
AGA02250
AGA02260
AGA02270
AGA02280
AGA02290
AGA02300
AGA02310
AGA02320
AGA02330
AGA02340
AGA02350
AGA02360
AGA02370
AGA02380
AGA02390
AGA02400
AGA02410
AGA02420
AGA02430
AGA02440
AGA02450
AGA02460
AGA02470
AGA02480
AGA02490
AGA02500
AGA02510
AGA02520
AGA02530
AGA02540
AGA02550
AGA02560
AGA02570
AGA02580
AGA02590
AGA02600
AGA02610
AGA02620
AGA02630
AGA02640
AGA02650

```

```

52 VOL=VOL+2.0944E0*(F(J)*R(J)**3.E0+F(J-1)*R(J-1)**3.E0)*(R(J)-R(J-1))
1 WRITE (IOUT,53) VOL
C THE VOLUME PER PARTICLE CALCULATED HERE IS OBTAINED USING
C ALL AVAILABLE (NMAX VALUES) VALUES FOR THE PARTICLE RADII.
WRITE (IOUT,54)
DO 56 INT=1,N1
INF=INT+1
56 WRITE (IOUT,55) INT,RR(INT),RR(INF)
1 FORMAT (6E12.6,I3)
2 FORMAT (1H, //24H AEROSOL PARAMETERS ARE )
3 FORMAT (1H, 24X, 6HRBAR= ,E12.6,5X, 7HSIGMA= ,E12.6, 7H RLO = ,E12.6,
+ 7H RHI = ,E12.6/)
4 FORMAT (1H, 24X, 5HRLO= ,E10.4, 1X, 5HRHI= ,E10.4, 1X, 5HCUE= ,E10.4,
+ 1X, 3HA= ,E10.4, 1X, 3HB= ,E10.4/)
5 FORMAT (1H, 24X, 5HRLO= ,E10.4, 1X, 5HRHI= ,E10.4, 1X, 5HCUE= ,E10.4,
+ 1X, 3HA= ,E10.4, 1X, 4HVIS= ,E10.4/)
6 FORMAT (1H, 24X, 5HRLO= ,E10.4, 1X, 5HRHI= ,E10.4, 1X, 4HRC= ,E10.4, 1X,
+ 5HALF= ,E10.4, 1X, 5HGAM= ,E10.4/)
7 FORMAT (1H, 5(26H RADIUS RELATIVE NO. )))
8 FORMAT (1H, 46H NORMALIZATION FACTOR FOR SIZE DISTRIBUTION = ,E14.7)
11 FORMAT (1X, 10(1PE12.6, 1X))
25 FORMAT (1X, 7HN(A) = ,E12.6, 2X, 9H RBARA = ,E12.6, 2X, 12H SIGMA(A) =
+ ,E12.6, 1X, 7HN(C) = ,E12.6, 2X, 9H RBARC = ,E12.6, 2X,
+ 12H SIGMA(C) = ,E12.6/)
27 FORMAT (1X, 42H MARSHALL-PALMER RAIN MODEL : RAIN RATE = ,1PE10.3,
+ 21H MM PER HOUR, DENS = ,1PE12.6, 8H PART/CC)
45 FORMAT (1H, 50H*** BIMODAL DISTRIBUTION...EQUIVALENT DENSITY IS
+ ,1PE13.6, 18H PARTICLES PER CC,/)
51 FORMAT (1H, 45H AVERAGE ANALYTIC DRY VOLUME PER PARTICLE IS ,3X,
+ ,1PE12.6, 18H CUBIC MICROMETERS)
53 FORMAT (1X, 47H AVERAGE NUMERICAL DRY VOLUME IS
+ ,1PE12.6, 18H CUBIC MICROMETERS/)
54 FORMAT (1X, 10X, 35H SIZE-INTERVALS USED ARE AS FOLLOWS/)
55 FORMAT (1H, 14H INTERVAL NO. , I3, 5X, 7HRMIN = , F11.5, 5X, 8H RMAX =
+ , F11.5)
RETURN
END

```

```

SUBROUTINE AGXP2(RELHUM,CTSUM,CSSUM,CRSUM,TVOL,TMASS,DENS,CATTN,
1 TEMP,DELTA,NINDX,IW,OLSTAR,OM2,LLLL,IT,WAVE,EM,CAY,EMM,MORTE,
2 PFNZRO)
REAL KEXT,KEXTT,KEXOLD
N.B. FFF IS AN ALIS FOR ARRAY FF
COMMON /PT1/ F(513),R(513),DR(8),RR(9),FFF(514),
+NRADI,PI,IDSTP,NKG,NHALV,N1
COMMON /PT2/ PHH(65),PSUMTT(65),PGG(65),PSUM(65),PSUMT(65),
1P(65)
COMMON /IO/ IOIN,IOUT,NUNIT,IEO,NEOU
COMMON /AGXM/ C(65),W(65),OLT(65),JDIMCK(3)
IN THIS SUBROUTINE THE FOLLOWING CONVENTIONS ARE USED IN
PREFIXING VARIABLE NAMES:
THE LETTER C IS USED FOR CROSS-SECTIONS
THE LETTER Q IS USED FOR EFFICIENCY FACTORS
THE LETTER K IS USED FOR EXTINCTION COEF. PER UNIT PATH (KM)
THE LETTER T IS A SUFFIX FOR TOTAL VALUES
THE LETTER O IS A PREFIX FOR OMEGA SUB 1 AND 2 CALCULATIONS
FOR THE IDSTP=6 AND 12 CASES, DENS IS USED TO TRANSFER THE
LIQUID WATER CONTENT FROM THE MAIN PROGRAM TO THIS SUBROUTINE;
ELWC IS USED AS THE AEROSOL CONCENTRATION FOR THOSE CASES.
IF (IDSTP.EQ.6.OR.IDSTP.EQ.12) ELWC=DENS
PZRSMT=0.
OLSTAR=0.0E0
OM2=0.0E0
CTSUMT=0.0E+00
CSSUMT=0.0E+00
DENST=0.0E0
CRSUMT=0.0E+00
EMM=1.0E0
NLINE=0
BH=1.056E-3
FACTORS BH AND CH ARE USED IN SIZE ADJUSTMENTS
FH IS THE SATURATION RATIO
FH=RELHUM/100.0E0
CH=FM/(1.0E0-FH)
CONCT=0.0E0
KEXTT=0.0E0
CONVERT VOL PER PARTICLE RECEIVED FROM MAIN PROGRAM VIA VARIABLE
TVOL TO DRY VOLUME PER PARTICLE IN CUBIC CENTIMETERS
DRYVOL=TVOL*1.0E-12
TVOL=0.0E0
TMASS=0.0E0
DO 6 J=1,IT
PSUMTT(J)=0.0E0
PHH(J)=0.0E0
PGG(J)=0.0E0
6
CONVERT TEMP. TO KELVIN FOR SUBROUTINE WATER USAGE
TEMK=TEMP+273.16E0
C
SKIP SUBROUTINE WATER FOR THE IDSTP = 12 CASE, AND READ THE
OPTICAL DATA FOR RAIN AS EMUA,CAYA,ETC...NEEDED BECAUSE CASE
IDSTP=12 MAY BE AT WAVELENGTHS LONGER THAN FOUND IN ROUTINE
WATER.
C
IF (IDSTP.EQ.12) GO TO 8
SUBROUTINE WATER RETURNS INTERPOLATED VALUES FOR EMW, CAYW AND
RHOW AT WAVELENGTH = WAVE AND AT TEMPERATURE = TEMK (DEG K).
C
EMW IS REAL PART OF INDEX OF REFR FOR PURE WATER AT TEMP(DEG C).
C
CAYW IS IMAG. PART OF INDEX OF REFR. FOR PURE WATER;
C
CAYW, HERE IS POSITIVE, BUT TREATED AS NEGATIVE IN MIE-ROUTINE.
C
RHOW IS MASS DENSITY(GM/CC) AT TEMPERATURE = TEMP (DEG C).
CALL WATER(WAVE,EMW,CAYW,TEMP,RHOW)
WRITE (IOUT,9) EMW,CAYW,TEMP,RHOW
C
BEGIN LOOP OVER AEROSOL COMPONENTS INDEXED BY NK
9 DO 32 NK=1,NINDX
C
BYPASS READ OF EMA,CAYA,ETC. FOR IDSTP=6 CASE..USE WATER DATA
IF (IDSTP.NE.6) GO TO 10
EMA=EMW
CAYA=CAYW
RHOA=RHOW
CONC=ELWC

```

```

AGB00010
AGB00020
AGB00030
AGB00040
AGB00050
AGB00060
AGB00070
AGB00080
AGB00090
AGB00100
AGB00110
AGB00120
AGB00130
AGB00140
AGB00150
AGB00160
AGB00170
AGB00180
AGB00190
AGB00200
AGB00210
AGB00220
AGB00230
AGB00240
AGB00250
AGB00260
AGB00270
AGB00280
AGB00290
AGB00300
AGB00310
AGB00320
AGB00330
AGB00340
AGB00350
AGB00360
AGB00370
AGB00380
AGB00390
AGB00400
AGB00410
AGB00420
AGB00430
AGB00440
AGB00450
AGB00460
AGB00470
AGB00480
AGB00490
AGB00500
AGB00510
AGB00520
AGB00530
AGB00540
AGB00550
AGB00560
AGB00570
AGB00580
AGB00590
AGB00600
AGB00610
AGB00620
AGB00630
AGB00640
AGB00650
AGB00660
AGB00670
AGB00680
AGB00690
AGB00700

```

	EMUA=0.0E0	AGB00710
	GO TO 11	AGB00720
C	*** READ OPTICAL AND PHYSICAL DATA ***	AGB00730
10	READ (IOIN,2) EMA,CAYA,EMUA,RHOA,CONC	AGB00740
	IF (IW.EQ.0) EMA=EMW	AGB00750
	IF (IW.EQ.0) CAYA=CAYW	AGB00760
	IF (IDSTP.NE.12) GO TO 11	AGB00770
	EMUA=0.0E0	AGB00780
	RHOW=RHOA	AGB00790
	EMW=EMA	AGB00800
	CAYW=CAYA	AGB00810
11	IF (RHOA.LE.0.E0) RHOA=1.E0	AGB00820
	WRITE (IOUT,3) NK,EMA,CAYA,RHOA,EMUA,CONC	AGB00830
	IF (EMA.LT.1.E-30) GO TO 44	AGB00840
	BHT=BH*(298.E0/TEMK)	AGB00850
	IF (EMUA.LE.0.01) CH=0.0	AGB00860
	BC=BHT*CH	AGB00870
	A=1.E0+((RHOA/RHOW)*EMUA*CH)	AGB00880
	AC=A**(.1.E0/3.E0)	AGB00890
C	ADJUST EM,RHO AND CAY PER G. HANEL/ADVANCES IN GEOPHYS/1976	AGB00900
	RHO=RHOW+(RHOA-RHOW)/A	AGB00910
	EM=EMW+(EMA-EMW)/A	AGB00920
	CAY=CAYW+(CAYA-CAYW)/A	AGB00930
	CAY=CAY/EM	AGB00940
C	INITIALIZE QUANTITIES USED TO HOLD RUNNING SUMMATIONS OVER	AGB00950
C	RADII FOR THE CURRENT COMPONENT	AGB00960
	CTSUM=0.E0	AGB00970
	CSSUM=0.E0	AGB00980
	CRSUM=0.0E0	AGB00990
	VOL=0.0E0	AGB01000
	OL1SUM=0.0E0	AGB01010
	OL2SUM=0.E0	AGB01020
	PZRSUM=0.	AGB01030
	DO 13 J=1,IT	AGB01040
13	PSUM(J)=0.0E0	AGB01050
C	PRINT HEADER IF DETAILED MIE RESULTS ARE TO BE PRINTED	AGB01060
	IF (MORTE.EQ.12345) WRITE (IOUT,5)	AGB01070
C	BEGIN ACTUAL LOOP OVER RADIUS INTERVALS FOR THE CURRENT NK VALUE	AGB01080
C	THIS LOOP IS THE ONE IN WHICH THE MIE CALCULATIONS ARE CALLED	AGB01090
C	INTERVALS ARE INDEXED BY I. THERE ARE NI SUCH INTERVALS.	AGB01100
	DO 26 I=1,NI	AGB01110
	NRADI=2	AGB01120
	D=RR(I+1)-RR(I)	AGB01130
C	RIT IS THE ADJUSTED RADIUS FOR THE RELATIVE HUMIDITY TO BE USED	AGB01140
C	IN THIS PARTICULAR RUN OR PASS	AGB01150
	RIT=RR(I)*AC-(BC/AC)	AGB01160
	IF (RIT.LT.RR(I).OR.RR(I).LT.0.04E0) RIT=RR(I)	AGB01170
	ALPHA=2.E0*PI*RIT/WAVE	AGB01180
C	ROUTINE MIEGX DOES THE ACTUAL MIE CALCULATIONS.	AGB01190
C	NOTE THAT THE IMAG. PART OF THE REFRACTIVE INDEX (CAY) HAS BEEN	AGB01200
C	NORMALIZED THROUGH DIVISION BY THE REAL PART (EM) BEFORE ITS	AGB01210
C	VALUE IS PASSED TO THE MIE-ROUTINE.	AGB01220
C	MIEGX RETURNS THE EXTINCTION EFFICIENCY FACTOR AS QT	AGB01230
C	MIEGX RETURNS THE SCATTERING EFFICIENCY FACTOR AS QS	AGB01240
C	MIEGX RETURNS THE BACK-SCATTERING (RADAR) EFFIC. FACTOR AS QR	AGB01250
C	MIEGX RETURNS THE AVERAGE INTENSITY (I1+I2)/2 IN THE ARRAY P<>	AGB01260
C	AT ANGLES = ARCCOS(C<>), WHERE C<> IS SET-UP BY	AGB01270
C	SUBROUTINE GUSET OR ANGLE	AGB01280
C	MIEGX ALSO RETURNS THE 2-ND AND 3-RD LEGENDRE EXPANSION COEF.	AGB01290
C	(OMEGA SUB 1 AND OMEGA SUB 2) AS O1STAR AND O2STAR.	AGB01300
	EMD=(EM)	AGB01310
	CAYD=(CAY)	AGB01320
	ALPHAD=(ALPHA)	AGB01330
	CALL MIEGX(EMD,CAYD,ALPHAD,QT,QS,QR,P,O1STRD,O2STRD,	AGB01340
+C,IT,PFNZRO)		AGB01350
	EM=(EMD)	AGB01360
	CAY=(CAYD)	AGB01370
	ALPHA=(ALPHAD)	AGB01380
	QT=(QT)	AGB01390
	QS=(QS)	AGB01400


```

14  QR=(QRD)
    O1STAR=(O1STRD)
    O2STAR=(O2STRD)
    KK=1+(I-1)*NKG
    IF (MORTE.EQ.12345) WRITE (IOUT,4) RIT,RR(I),F(KK),ALPHA,QT,QS,QR
    FKK=F(KK)
    FKKA=FKK*PI*RIT**2.E0
    VOLHH=4.1888*FKK*RIT**3.E0
    OL1HH=O1STAR*FKKA*QT
    OL2HH=O2STAR*FKKA*QT
    CTHH=QT*FKKA
    CSHH=QS*FKKA
    CRHH=QR*FKKA
    DO 14 J=1,IT
    PHH(J)=P(J)*FKK
    CONTINUE
    PFNZER=PFNZRO*FKK
    RIT=RR(I+1)*AC-(BC/AC)
    IF (RIT.LT.RR(I+1).OR.RR(I+1).LT.0.04E0) RIT=RR(I+1)
    ALPHA=2.E0*PI*RIT/WAVE
    EMD=(EM)
    CAYD=(CAY)
    ALPHAD=(ALPHA)
    CALL MIEGX(EMD,CAYD,ALPHAD,QT,QSD,QRD,P,O1STRD,O2STRD,
    +C,IT,PFNZRO)
    EM=(EMD)
    CAY=(CAYD)
    ALPHA=(ALPHAD)
    QT=(QTD)
    QS=(QSD)
    QR=(QRD)
    O1STAR=(O1STRD)
    O2STAR=(O2STRD)
    KK1=1+NKG*I
    FKK1=F(KK1)
    IF (MORTE.EQ.12345) WRITE (IOUT,4) RIT,RR(I+1),F(KK1),ALPHA,QT,QS,QR
    FKK1A=F(KK1)*PI*RIT**2.E0
    VOLHH=(VOLHH+4.188E0*FKK1*RIT**3.E0)*D*0.5E0
    OL1HH=(OL1HH+F(KK1A)*QT*O1STAR)*D*0.5E0
    OL2HH=(OL2HH+F(KK1A)*QT*O2STAR)*D*0.5E0
    CTHH=(CTHH+QT*F(KK1A))*D*0.5E0
    CSHH=(CSHH+QS*F(KK1A))*D*0.5E0
    CRHH=(CRHH+QR*F(KK1A))*D*0.5E0
    DO 15 J=1,IT
    PHH(J)=(PHH(J)+P(J)*F(KK1))*D*0.5E0
    CONTINUE
    PFNZER=(PFNZER+PFNZRO*F(KK1))*D*0.5
    FF=0.5E0*D*(F(KK)+F(KK1))
    NT=1
    N=1
16  NJ=NT
    NT=2*NT
    D=0.5E0*D
    VOLGG=0.0E0
    OL1GG=0.0E0
    OL2GG=0.0E0
    CTGG=0.0E0
    CSGG=0.0E0
    CRGG=0.0E0
    FT=0.0E0
    DO 17 J=1,IT
    PGG(J)=0.0E0
    PZRTMP=0.
17  NEXT LOOP HANDLES INTERMEDIATE PARTICLE SIZES..THOSE LYING BETWEEN
    C RMIN AND RMAX FOR THE CURRENT INTERVAL WHOSE INDEX IS I,
    DO 19 JG=1,NJ
    KK=1+(I-1)*NKG+(2*JG-1)*(NKG/NT)
    RIT=R(KK)*AC-(BC/AC)
    IF (RIT.LT.R(KK).OR.R(KK).LT.0.04E0) RIT=R(KK)
    ALPHA=2.E0*PI*RIT/WAVE

```

```

AGB001410
AGB001420
AGB001430
AGB001440
AGB001450
AGB001460
AGB001470
AGB001480
AGB001490
AGB001500
AGB001510
AGB001520
AGB001530
AGB001540
AGB001550
AGB001560
AGB001570
AGB001580
AGB001590
AGB001600
AGB001610
AGB001620
AGB001630
AGB001640
AGB001650
AGB001660
AGB001670
AGB001680
AGB001690
AGB001700
AGB001710
AGB001720
AGB001730
AGB001740
AGB001750
AGB001760
AGB001770
AGB001780
AGB001790
AGB001800
AGB001810
AGB001820
AGB001830
AGB001840
AGB001850
AGB001860
AGB001870
AGB001880
AGB001890
AGB001900
AGB001910
AGB001920
AGB001930
AGB001940
AGB001950
AGB001960
AGB001970
AGB001980
AGB001990
AGB002000
AGB002010
AGB002020
AGB002030
AGB002040
AGB002050
AGB002060
AGB002070
AGB002080
AGB002090
AGB002100

```

	EMD= (EM)	AGB02110
	CAYD= (CAY)	AGB02120
	ALPHAD= (ALPHA)	AGB02130
	CALL MIEGX(EMD, CAYD, ALPHAD, QTD, QSD, QRD, P, O1STRD, O2STRD,	AGB02140
	+C, IT, PFNZRO)	AGB02150
	EM= (EMD)	AGB02160
	CAY= (CAYD)	AGB02170
	ALPHA= (ALPHAD)	AGB02180
	QT= (QTD)	AGB02190
	QS= (QSD)	AGB02200
	QR= (QRD)	AGB02210
	O1STAR= (O1STRD)	AGB02220
	O2STAR= (O2STRD)	AGB02230
	IF (MORTE.EQ.12345) WRITE (IOUT,4) RIT,R(KK),F(KK),ALPHA,QT,QS,QR	AGB02240
	NRADI=NRADI+1	AGB02250
	FKK=F(KK)	AGB02260
	FKKA=FKK*PI*RIT**2.E0	AGB02270
	VOLGG=4.1888E0*FKK*RIT**3.0E0+VOLGG	AGB02280
	OL1GG=OL1GG+O1STAR*FKKA*QT	AGB02290
	OL2GG=OL2GG+O2STAR*FKKA*QT	AGB02300
	CTGG=CTGG+QT*FKKA	AGB02310
	CSGG=CSGG+QS*FKKA	AGB02320
	CRGG=CRGG+QR*FKKA	AGB02330
	DO 18 J=1,IT	AGB02340
	PGG(J)=PGG(J)+P(J)*FKK	AGB02350
18	CONTINUE	AGB02360
	PZRTMP=PZRTMP+PFNZRO*FKK	AGB02370
19	FT=FT+FKK	AGB02380
C	ADD RESULTS ACCUMULATED DURING PREVIOUS HALVINGS TO THOSE FOUND	AGB02390
C	FOR THE NEW RADII TREATED WITHIN THE LOOP OVER INDEX JG	AGB02400
	VOLHHT=0.5E0*VOLHH+D*VOLGG	AGB02410
	OL1HHT=0.5E0*OL1HH+D*OL1GG	AGB02420
	OL2HHT=0.5E0*OL2HH+D*OL2GG	AGB02430
	CTHHT=0.5E0*CTHH+D*CTGG	AGB02440
	CSHHT=0.5E0*CSHH+D*CSGG	AGB02450
	CRHHT=0.5E0*CRHH+D*CRGG	AGB02460
	DO 20 J=1,IT	AGB02470
20	PHH(J)=.5E0*PHH(J)+D*PGG(J)	AGB02480
	PFNZER=.5E0*PFNZER+D*PZRTMP	AGB02490
	FFT=0.5E0*FF+D*FT	AGB02500
	IF (CTHHT.LT.1.E-30) GO TO 22	AGB02510
	DEL=ABS(VOLHHT-VOLHH)/ABS(VOLHHT)	AGB02520
	IF (DEL.LE.DELTA) GO TO 21	AGB02530
	GO TO 22	AGB02540
21	IF (N.GT.2) GO TO 24	AGB02550
C	DO NOT ALLOW DEL LESS THAN DELTA EXIT UNLESS AT LEAST TWO	AGB02560
C	HAVINGS HAVE BEEN DONE	AGB02570
22	IF (N.EQ.NHALV) GO TO 24	AGB02580
C	MUST EXIT WHEN NHALV HALVINGS HAVE BEEN DONE EVEN IF THE DELTA	AGB02590
C	CRITERION HAS NOT BEEN SATISFIED..SINCE NO MORE VALUES OF RADII	AGB02600
C	ARE AVAILABLE.	AGB02610
	FF=FFT	AGB02620
	CRHH=CRHHT	AGB02630
	OL1HH=OL1HHT	AGB02640
	OL2HH=OL2HHT	AGB02650
	CSHH=CSHHT	AGB02660
	CTHH=CTHHT	AGB02670
	VOLHH=VOLHHT	AGB02680
	N=N+1	AGB02690
24	GO TO 16	AGB02700
	CONTINUE	AGB02710
	IF(N.EQ.NHALV) WRITE(IOUT,124) I	AGB02720
C	SUM QUANTITIES OVER ALL INTERVALS TREATED UP UNTIL NOW	AGB02730
	CTSUM=CTSUM+CTHHT	AGB02740
	CSUM=CSUM+CSHHT	AGB02750
	CRSUM=CRSUM+CRHHT	AGB02760
	VOL=VOL+VOLHHT	AGB02770
	OL1SUM=OL1SUM+OL1HHT	AGB02780
	OL2SUM=OL2SUM+OL2HHT	AGB02790
C	AT THIS POINT, PSUM() IS THE RUNNING SUM OF THE AVG. INTENSITY	AGB02800

C	AS SUMMED OVER SIZES	AGB02810
	DO 25 J=1,IT	AGB02820
25	PSUM(J)=PSUM(J)+PMH(J)	AGB02830
	PZRSUM=PZRSUM+PFNZER	AGB02840
	WRITE (IOUT,27) NK,I,NRADI,CTHHT	AGB02850
	NLINES=NLINES+NRADI	AGB02860
C	END LOOP OVER HALVING INTERVALS INDEXED BY I	AGB02870
26	CONTINUE	AGB02880
C	CALCULATE PARTICLE NUMBER DENSITY<NO. PER CC> AS DENSC	AGB02890
	DENSC=CONC/(RHOA*DRYVOL)	AGB02900
C	OVERRIDE CALCULATED VALUE OF DENSC WITH DENS IF LLLL =1	AGB02910
	IF (LLLL.EQ.1) DENSC=DENS	AGB02920
C	RECALCULATE CONC FROM OTHER INPUT DATA IF LLLL=1	AGB02930
	IF (LLLL.EQ.1) CONC=DENS*RHOA*DRYVOL	AGB02940
C	REPLACE DENS BY DENSC FOR LATER USE BY AGXP3	AGB02950
	DENS=DENSC	AGB02960
C	WEIGHT CTSUM,ETC. BY NUMBER DENSITIES (DENSC) FOR THIS COMPONENT	AGB02970
	CTSUM=CTSUM+DENSC	AGB02980
	CSSUM=CSSUM+DENSC	AGB02990
	CRSUM=CRSUM+DENSC	AGB03000
	VOL=VOL+DENSC	AGB03010
	OL1SUM=OL1SUM+DENSC	AGB03020
	OL2SUM=OL2SUM+DENSC	AGB03030
	DO 29 J=1,IT	AGB03040
29	PSUM(J)=PSUM(J)+DENSC	AGB03050
	PZRSUM=PZRSUM+DENSC	AGB03060
C	NOW, SUM OVER COMPONENTS INDEXED BY NK	AGB03070
C	CONCT IS THE TOTAL DRY-AEROSOL CONCENTRATION IN MG PER CC	AGB03080
	CONCT=CONCT+1.E3*CONC	AGB03090
	DENST=DENST+DENSC	AGB03100
	OLSTAR=OL1SUM+OLSTAR	AGB03110
	OM2=OL2SUM+OM2	AGB03120
	CTSUMT=CTSUMT+CTSUM	AGB03130
C	AT THIS POINT, CTSUMT IS THE TOTAL EXTINCTION CROSS SECTION	AGB03140
CC	(IN SQ. MICRONS) AS SUMMED OVER ALL COMPONENTS WHICH	AGB03150
C	HAVE BEEN DEALT WITH THUS FAR	AGB03160
	CSSUMT=CSSUMT+CSSUM	AGB03170
	CRSUMT=CRSUMT+CRSUM	AGB03180
	DO 30 J=1,IT	AGB03190
30	PSUMTT(J)=PSUM(J)+PSUMTT(J)	AGB03200
	PZRSMT=PZRSMT+PZRSUM	AGB03210
	VOL=VOL*1.E-12	AGB03220
CC	TVOL IS THE TOTAL VOLUME (IN CM**3) OCCUPIED BY THE AEROSOL	AGB03230
C	PARTICLES. TVOL IS NOT ACTUALLY USED IN THIS VERSION OF	AGB03240
	PROGRAM AGAUS.	AGB03250
	TVOL=VOL+TVOL	AGB03260
	EMASS=VOL*RHO	AGB03270
	TMASS=TMASS+EMASS	AGB03280
	KEXOLD=KEXTT	AGB03290
	KEXTT=CTSUMT*1.E-3	AGB03300
	KEXT=KEXTT-KEXOLD	AGB03310
	WRITE (IOUT,31) NK,VOL,EMASS,KEXT	AGB03320
C	VPF(VOL) IS THE VOLUME PACKING FRACTION; THAT IS, THE FRACTION	AGB03330
CC	OF EACH CC OF SPACE WHICH IS FILLED BY AEROSOL MATERIAL BELONGING	AGB03340
C	TO THE CURRENT COMPONENT NK.	AGB03350
CC	TMASS IS THE TOTAL MASS OF AEROSOL FOUND IN 1 CC OF SPACE.	AGB03360
CC	EMASS IS THE MASS OF AEROSOL MATERIAL PER CC ASSOCIATED WITH	AGB03370
CC	THE CURRENT COMPONENT NK.	AGB03380
CC	KEXT IS THE EXTINCTION COEF.(PER KM) WHICH IS ASSOCIATED WITH	AGB03390
C	THE CURRENT COMPONENT--AS IF IT ALONE WERE PRESENT.	AGB03400
CC	KEXTT IS THE SUM OF THE KEXT'S OVER ALL COMPONENTS.	AGB03410
C	END LOOP OVER AEROSOL COMPONENTS INDEXED BY NK.	AGB03420
	IF (NINDX.GT.1) WRITE (IOUT,42) NK	AGB03430
	IF (MORTE.EQ.12345) WRITE (IOUT,43)	AGB03440
32	CONTINUE	AGB03450
	IF (NINDX.GT.1) WRITE (IOUT,33) TMASS,KEXTT	AGB03460
	WRITE (IOUT,34) NLINES	AGB03470
	DENS=DENST	AGB03480
	NRADI=NLINES	AGB03490
C	NOW, PERFORM THE FINAL RENORMALIZATIONS TO OBTAIN CTSUM, ETC.	AGB03500

```

C      VALUES REPRESENTATIVE OF A SINGLE AVERAGE PARTICLE.
C      CTSUM BECOMES THE EXTINCTION CROSSSECTION IN SQ. MICROMETERS PER
C      AVERAGE PARTICLE. THE OTHER QUANTITIES CARRY SIMILAR MEANINGS.
      DO 35 J=1,IT
35     PSUM(J)=PSUMTT(J)/DENST
      OLSTAR=OLSTAR/CTSUMT
      OM2=OM2/CTSUMT
      CTSUM=CTSUMT/DENST
      CSSUM=CSSUMT/DENST
      CRSUM=CRSUMT/DENST
C      ... AND CONVERT ANG INTENS AT ZERO DEGREES INTO PHASE FUNCT.
C      WITH NORMALIZATION
      PFNZRO=PZRSMT/DENST*(WAVE*WAVE/(PI*CTSUM*EMM*EMM))
      WRITE(IOUT,36) OLSTAR,OM2,PFNZRO
C      CALCULATE ATTENUATION COEFS. IN SQ.METERS PER MILLIGRAM
      CATTN=CTSUMT*1.E-12/CONCT
      CATTNW=1.E-12*KEXTT/TMASS
      WRITE(IOUT,37) CATTN
      IF(RELHUM.GT.1.0)WRITE(IOUT,38) CATTNW
      GO TO 41
44     WRITE(IOUT,45) EMA
      STOP
      2 FORMAT(4F10.6,E15.7)
      3 FORMAT(1H,6H INDX=,I3,4H M=,F10.6,6H K=-,F10.6,9H1. MASS,
      +      10H DENSITY=,F8.6,
      +      17H GROWTH FACTOR=,F8.4,9H. CONC=,1PE12.5,7H GM/CC/)
      4 FORMAT(1X,F10.5,6(2X,1PE11.5))
      5 FORMAT(/54H R(MICRONS) DRY RADIUS N(R) MIE SIZE
      +      36HQ(EXT) Q(SCA) Q(RADAR)/)
      9 FORMAT(1H,39H INDEX OF REFRACTION FOR PURE WATER IS: ,F8.6,
      +      3H -,F8.6,1H1//,1X,25H MASS DENSITY OF WATER AT ,F6.2,
      +      11H DEG C IS: ,F8.2,6H GM/CC./)
124    FORMAT(/52H *** CONVERGENCE LEVEL NOT REACHED FOR INTERVAL NO. ,
      +      I3,4H ***/)
      27 FORMAT(1H,19H FOR COMPONENT NO. ,I3,15H INTERVAL NO. ,I3,1H,14,
      +      43H RADII WERE USED. CONTRIBUTION TO CTSUM = ,1PE12.6)
      31 FORMAT(1H,7,20H FOR COMPONENT NO. ,I2,12H: VPF = ,1PE12.5,
      +      7H PER CC,24H MASS CONCENTRATION = ,E12.5,21H GM/CC. KEXT =
      +      E12.5,7H PER KM)
      33 FORMAT(1H,7,29H TOTAL MASS CONCENTRATION = ,1PE12.5,7H GM/CC.,
      +      15H TOTAL KEXT = ,E12.5,7H PER KM)
      34 FORMAT(/1X,32H TOTAL NUMBER OF RADII USED WAS ,I5)
      36 FORMAT(/1X,19H ANALYTIC SOLUTIONS,/,
      +      16H OMEGA SUB 1 = ,1PE14.7/,16H OMEGA SUB 2 = ,1PE14.7/
      +      16H PFN AT ZERO = ,1PE14.7/)
      37 FORMAT(1H,21H ATTENUATION COEF. = ,1PE12.5,12H SQ-METERS/,
      +      33H MILLIGRAM OF DRY AEROSOL MATERIAL)
      38 FORMAT(1H,21H ATTENUATION COEF. = ,1PE12.5,12H SQ-METERS/,
      +      33H MILLIGRAM OF WET AEROSOL MATERIAL/)
      42 FORMAT(1H,7,1X,10X,30(1H*),33H END OF AEROSOL COMPONENT CYCLE,
      +      7H NUMBER ,I3,2X,30(1H*)//)
      43 FORMAT(1H)
      45 FORMAT(/1X,11H***** EMA(,F10.6,20H) IS EITHER ZERO OR ,
      +      35H NEGATIVE - PROGRAM TERMINATED *****)
41     RETURN
      END

```

AGB03510
 AGB03520
 AG 03530
 AGB03540
 AGB03550
 AGB03560
 AGB03570
 AGB03580
 AGB03590
 AGB03600
 AGB03610
 AGB03620
 AGB03630
 AGB03640
 AGB03650
 AGB03660
 AGB03670
 AGB03680
 AGB03690
 AGB03700
 AGB03710
 AGB03720
 AGB03730
 AGB03740
 AGB03750
 AGB03760
 AGB03770
 AGB03780
 AGB03790
 AGB03800
 AGB03810
 AGB03820
 AGB03830
 AGB03840
 AGB03850
 AGB03860
 AGB03870
 AGB03880
 AGB03890
 AGB03900
 AGB03910
 AGB03920
 AGB03930
 AGB03940
 AGB03950
 AGB03960
 AGB03970
 AGB03980
 AGB03990
 AGB04000
 AGB04010
 AGB04020
 AGB04030
 AGB04040
 AGB04050
 AGB04060

```

SUBROUTINE AGXP3(CTSUM,CSSUM,CRSUM,GNU,DENS,NINDX,
+ WAVE,EM,CAY,EMM,IT,IEND,IANG)
COMMON /PT2/ PC(65),OL(65),RMS(65),PSUM(65),PSUMT(65),P(65)
COMMON /IO/ IOIN,IOUT,NUNIT,IEO,NEOU
COMMON /AGXM/ C(65),W(65),OLT(65),JDIMCK(3)
COMMON /PT1/ F(513),R(513),DR(8),RR(9),FF(514)
+ ,NRADI,PI,IDSTP,NKG,NHALV,NI
C IEND=1 WHEN THE COMPOSITE PHASE FUNCTION IS BEING WRITTEN
IF (IEND.EQ.1) GO TO 6
ALBDO=CSSUM/CTSUM
PFACT IS USED TO CONVERT AVG. INTENSITY PSUM( ) INTO PHASE-
FUNCTIONS. SFACT IS USED TO CONVERT PSUM INTO SCATTERING
FRACTIONS, NORMALIZED PER SOM. THE INTEGRAL OF SCAT OVER SOLID
ANGLE SHOULD YIELD THE TOTAL SCATTERING CROSS-SECTION IN SQ. M.
SFACT=WAVE*WAVE*DENS*1.E-6/(4.*PI*PI)
PFACT=WAVE*WAVE/(PI*CTSUM*EMM*EMM)
DO 15 J=1,IT
SCAT=PSUM(J)*SFACT
PSUM(J)=PSUM(J)*PFACT
C UNCOMMENT THE FOLLOWING STMT IF U WANT SCAT FRACT,COSINES AND A
C COUNTER WRITTEN ON NUNIT. NUNIT IS SET IN THE BLOCK DATA ROUTINE.
C WRITE (NUNIT,3) SCAT,C(J),J
15 CONTINUE
IF (NINDX.GE.2) WRITE (IOUT,12)
WRITE (IOUT,4) IDSTP,WAVE,EM,CAY,CTSUM,CSSUM,ALBDO
C CONVERT AVG. CROSS-SECTIONS TO COEFFICIENTS (PER KM)
CTSUM=CTSUM*1.0E-3*DENS
CSSUM=CSSUM*1.0E-3*DENS
CRSUM=CRSUM*1.0E-3*DENS
WRITE (IOUT,13) CTSUM,CSSUM,CRSUM
WRITE (IOUT,14) GNU,DENS
IF (IT.LT.2) GO TO 21
C WRITE PHASE FUNCTION AT SPECIFIED ANGLES
6 WRITE (IOUT,1)
WRITE (IOUT,5)
C FIND ANGLES FROM COSINES
DO 2 I=1,IT
2 FF(I)=180.*ATAN2(SQRT(1.-C(I)**2),C(I))/PI
DO 19 J=1,IT,4
K=J+3
IF (K.GT.IT) K=IT
19 WRITE (IOUT,8) (C(I),FF(I),PSUM(I),I=J,K)
IF (IANG.NE.0) RETURN
C ROUTINE GAUS GENERATES AND PRINTS THE LEGENDRE
C EXPANSION COEFS (OMEGAS) FOR THE PHASE FUNCTION.
CALL GAUS(IT)
C CHECK TO SEE IF SNG. SCAT. ALBEDO (ALBDO) COMPUTED DIRECTLY
C FROM CROSS-SECTIONS AGREES WITH THAT FOUND FROM THE LEGENDRE
C EXPANSION OF PHASE-FUNCTION.
IF ((ABS(OL(1)-ALBDO)/ALBDO.GT.5.E-3).AND.(IEND.NE.1))
1 WRITE (IOUT,20)
1 FORMAT (//1H,50X,14H PHASE FUNCTION/,1X,42X,31H(NORMALIZED TO 4 PI
+ OMEGA ZERO)/)
3 FORMAT (2(E13.7,1X),I3)
4 FORMAT (1H1,/,41H DISTRIBUTION WAVELENGTH REFRACTIVE,9X,
+ 20H EXTINCTION X SECTION,8X,20H SCATTERING X SECTION,12X,5H ALBDO/,
+ 1H,6X,4H TYPE,6X,9H(MICRONS),8X,5H INDEX,16X,12H(SQ MICRONS),13X,
+ 12H(SQ MICRONS)/1H,19,4X,F11.4,F10.4,3H(1-,F7.4,2H1),
+ 7X,1PE14.7,11X,1PE14.7,12X,1PE14.7,/)
5 FORMAT (1H,3X,4(5H MU,2X,ANGLE,17H PHASE FUNCTION ))
8 FORMAT (1H,F9.5,F7.2,E12.5,3(3X,F9.5,F7.2,E12.5))
12 FORMAT (52H THIS IS A MIXED CASE * SUBSEQUENT REFRACTIVE INDEX ,
+ 34H PRINT-OUTS ARE NOT GENERALLY VALID./)
13 FORMAT (1H,10H K(EXT) = ,1PE13.7,11H; K(SCA) = ,E13.7,
+ 11H; K(RAD) = ,E13.7,11H ALL PER KM/)
14 FORMAT (//14H WAVENUMBER = ,1PE12.6,5H CM-1,5X,10H DENSITY = ,E12.6,
+ 17H PARTICLES PER CC/)
20 FORMAT (//12H *** VALUES
+ ,55H OF ALBDO AND OL(1) DISAGREE BY MORE THAN 0.5 PERCENT ***
+ ,34H LARGER VALUE OF 'IT' IS NEEDED ***/)

```

21

RETURN
END

AGC00710
AGC00720

C	SUBROUTINE ANGLE(PI, IANG, IT)	ANG00010
C	THIS ROUTINE IS TO BE USED TO REPLACE GUSSET FOR THE PURPOSE	ANG00020
C	OF USING AGAUS TO DO PHASE FUNCTION CALCULATIONS AT -IT-	ANG00030
C	ANGLES BETWEEN 0 AND 180 DEGREES, RATHER THAN AT THE G-L	ANG00040
C	QUADRATURE ABSCISSA VALUES. IT ALSO READS THE INPUT ANGLES	ANG00050
C	IF IANG=2.	ANG00060
	COMMON /IO/ IOIN, IOOUT, NUNIT, IE0, NEOU	ANG00070
	COMMON /AGXM/ C(65), W(65), OLT(65), JDIMCK(3)	ANG00080
	RADS=PI/180.	ANG00090
	DEL=180./FLOAT(IT-1)	ANG00100
	IF (IE0.EQ.1.OR.IE0.EQ.2.OR.IE0.EQ.5) GO TO 4	ANG00110
	IF (IANG.EQ.2) GO TO 2	ANG00120
	DO 1 I=1, IT	ANG00130
	W(I)=DEL*FLOAT(I-1)	ANG00140
1	C(I)=COS(W(I)*RADS)	ANG00150
	RETURN	ANG00160
2	READ(5, 100) (W(I), I=1, IT)	ANG00170
4	DO 3 I=1, IT	ANG00180
3	C(I)=COS(W(I)*RADS)	ANG00190
	IANG=1	ANG00200
	RETURN	ANG00210
100	FORMAT(16F5.1)	ANG00220
	END	ANG00230

	FUNCTION GAMMA(X)	GMA00010
C	GAMMA FUNCTION: TAKEN FROM HANDBOOK OF MATHEMATICAL FUNCTIONS,	GMA00020
C	ABRAMOWITZ AND STEGUN, NOV 1964, PP 256-257. RECURRENCE FORMULA	GMA00030
C	6.1.16, POLYNOMIAL APPROXIMATION 6.1.35.	GMA00040
	DATA A1,A2,A3,A4,A5	GMA00050
	+ /- .5748646, .9512363, -.6998588, .4245549, -.1010678/	GMA00060
C	COMPUTER AND GAMMA FUNCTION LIMITS	GMA00070
	IF (X.GT.34..OR.X.LT.0.) GO TO 3	GMA00080
	GSUM=1	GMA00090
	N=IFIX(X+.00001)	GMA00100
C	FIND Z, LE. 1.	GMA00110
	Z=X-FLOAT(N)	GMA00120
C	OK FOR Z BEING INTEGER	GMA00130
	IF (Z.LT.1.E-04) N=N-1	GMA00140
	IF (Z.LT.1.E-04) Z=1.	GMA00150
C	COMPENSATE FOR N-1 IN FORMULA	GMA00160
	N=N-1	GMA00170
C	IF Z, LE. 1, SKIP LOOP	GMA00180
	IF (N.LE.0) GO TO 2	GMA00190
C	RECURRENCE RELATION: G(N+Z)=(N-1+Z)*(N-2+Z)...(1+Z)*G(1+Z)	GMA00200
	DO 1 I=1,N	GMA00210
	VALUE=FLOAT(I)+Z	GMA00220
1	GSUM=GSUM*VALUE	GMA00230
C	POLYNOMIAL APPROXIMATION: Z, LE. 1	GMA00240
2	GAMMA=1.+A1*Z+A2*Z*Z+A3*Z*Z*Z+A4*Z*Z*Z*Z+A5*Z*Z*Z*Z*Z	GMA00250
	GAMMA=GAMMA*GSUM	GMA00260
	RETURN	GMA00270
3	WRITE (1,100) X	GMA00280
100	FORMAT(1H, ' ***** THE VALUE OF X (', 2PE11.4, ') IS EITHER ',	GMA00290
	+ ' OUTSIDE COMPUTER LIMITS', /, ' OR NEGATIVE - PGM STOPPED *****')	GMA00300
	STOP	GMA00310
	END	GMA00320

	SUBROUTINE GUSET(IT)	GUSET010
	THIS ROUTINE CALCULATES THE ABSCISSAE C()	GUSET020
C	AND GAUSS-LEGENDRE WEIGHTS W()	GUSET030
C	VIA GAUSS-LEGENDRE QUADRATURE OF ORDER N	GUSET040
	COMMON /IO/ IOIN, IOOUT, NUNIT, IEO, NEOU	GUSET050
	COMMON /AGXM/ C(65), W(65), OLT(65), JDIMCK(3)	GUSET060
	COMMON /PT1/ F(513), R(513), DR(8), RR(9), FF(514)	GUSET070
	+ ,NRADI, PI, IDSTP, NKG, NHALV, NI	GUSET080
	N=IT	GUSET090
	TOL=1.0E-06	GUSET100
	AA=2.0E+00/PI**2.0E+00	GUSET110
	AB=-62.0E+00/(3.0E+00*PI**4.0E+00)	GUSET120
	AC=15116.0E+00/(15.0E+00*PI*6.0E+00)	GUSET130
	AD=-12554474.0E+00/(105.0E+00*PI*8.0E+00)	GUSET140
	F(1)=1.0E+00	GUSET150
	EN= FLOAT(N)	GUSET160
	NP1=N+1	GUSET170
	U=1.0E+00-(2.0E+00/PI)**2.0E+00	GUSET180
	D=1.0E+00/SQRT((EN+0.5E+00)**2.0E+00+U/4.0E+00)	GUSET190
	DO 1 I=1, N	GUSET200
	S= FLOAT(I)	GUSET210
	A=4.0E+00*S-1.0E+00	GUSET220
	AE=AA/A	GUSET230
	AF=AB/A**3.0E+00	GUSET240
	AG=AC/A**5.0E+00	GUSET250
	AH=AD/A**7.0E+00	GUSET260
1	R(1)=PI*(A+AE+AF+AG+AH)/4.0E+00	GUSET270
	DO 6 K=1, N	GUSET280
	X=COS(R(K)*D)	GUSET290
2	F(2)=X	GUSET300
	DO 3 NN=3, NP1	GUSET310
	ENN= FLOAT(NN-1)	GUSET320
3	F(NN)=((2.0E+00*ENN-1.0E+00)*X*F(NN-1)-(ENN-1.0E+00)*F(NN-2))/ENN	GUSET330
	IF (ABS(F(NN)).GT.1E+35) F(NN)=SIGN(1.E+35, F(NN))	GUSET340
	PNP=EN*(F(N)-X*F(NP1))/(1.0E+00-X*X)	GUSET350
	XI=X-F(NP1)/PNP	GUSET360
	XD= ABS(XI-X)	GUSET370
	XDD=XD-TOL	GUSET380
	IF (XDD) 5,5,4	GUSET390
4	X=XI	GUSET400
	GO TO 2	GUSET410
5	C(K)=X	GUSET420
6	W(K)=2.0E+00*(1.0E+00-X*X)/(EN*F(N)*EN*F(N))	GUSET430
	DO 7 I=1, N	GUSET440
	R(I)=0.00	GUSET450
7	F(I)=0.00	GUSET460
	RETURN	GUSET470
	END	GUSET480

```

SUBROUTINE MIEGX(EMD,CAYD,ALPHAD,QTD,QSD,QRD,P,O1STRD,O2STRD,
+C,IT,PFNZRO)
THIS ROUTINE IS CURRENTLY SINGLE PRECISION COMPLEX
C CHANGE THE VALUE OF NDIM IF YOU CHANGE THE DIM OF A IN NEXT LINE
C COMPLEX A(600),ACAPN,ZNUM,ZDEN,ZPDT,ZRPDT,ZAN,ZANP,Y,RF,RRF,
1 RRF,X,WM1,FNA,FNB,TC1,FNAP,FNBP,FNAPP,FNBPP,TC2,WFN(2)
COMMON /IO/ IOIN,IOUT,NUNIT,IEO,NEOU
DIMENSION P(65),C(65)
DIMENSION T(4),TA(4),TB(2),TC(2),TD(2),TE(2),TF(2),TG(2)
DIMENSION ELTRMX(4,76),PI(3,76),TAU(3,76)
EQUIVALENCE (WFN(1),TA(1)),(FNA,TB(1)),(FNB,TC(1)),(FNAP,TD(1))
EQUIVALENCE (FNBP,TE(1)),(FNAPP,TF(1)),(FNBPP,TG(1))
C THESE EQUIVALENCES ALLOW USE OF REAL AND IMAG PARTS INDIVIDUALLY
TOL = 1.E-06
ITT=IT
X=(ALPHAD)
CAY=(CAYD)
EM=(EMD)
CAYE=CAY*EM
QRT=0.0
S=1.0
RF=CMPLX(EM,-CAYE)
NMX=IFIX(X*(EM+CAYE))+9
RRF=1.0/RF
RX=1.0/X
RRFX=RRF*RX
C LOOP POINT FOR CALCULATING PFN AT ZERO DEGREES
IAPXCT=0
21 CONTINUE
C THESE ARE THE PI AND TAU FUNCTIONS
DO 1 J=1,IT
PI(1,J)=0.0
PI(2,J)=1.0
TAU(1,J)=0.0
TAU(2,J)=C(J)
1 CONTINUE
T(1)=COS(X)
T(2)=SIN(X)
WM1=CMPLX(T(1),-T(2))
WFN(1)=CMPLX(T(2),T(1))
WFN(2)=RX*WFN(1)-WM1
T(1)=CAYE*X
N=1
C NDIM MUST EQUAL THE DIMENSION OF A( ).
NDIM=600
IF (NMX.LT.NDIM) NDELTA=NMX
IF (NMX.GT.NDIM) NDELTA=NDIM
NMX=0
IF (N.EQ.1) GO TO 4
EN=FLOAT(N)
T(1)=2.0*EN-1.0
T(2)=EN-1.0
T(3)=2.0*EN+1.0
DO 3 J=1,IT
PI1J=PI(1,J)
PI2J=PI(2,J)
CJ= C(J)
C SWITCH FOR CALCULATING PFN AT ZERO DEGREES
IF (IAPXCT.EQ.1) CJ=1.0
S2T=(1.0-CJ*CJ)
PI(3,J)=(T(1)*PI2J*CJ-EN*PI1J)/T(2)
TAU(3,J)=CJ*(PI(3,J)-PI1J)-T(1)*S2T*PI2J+TAU(1,J)
3 CONTINUE
WM1=WFN(1)
WFN(1)=WFN(2)
WFN(2)=T(1)*RX*WFN(1)-WM1
4 CONTINUE
C CALCULATE RATIO OF BESSEL FNS OF CONSECUTIVE ORDER
IF (N.LT.(NMX+1)) GO TO 9
NMX=NMX+NDELTA

```

```

MIEGX010
MIEGX020
MIEGX030
MIEGX040
MIEGX050
MIEGX060
MIEGX070
MIEGX080
MIEGX090
MIEGX100
MIEGX110
MIEGX120
MIEGX130
MIEGX140
MIEGX150
MIEGX160
MIEGX170
MIEGX180
MIEGX190
MIEGX200
MIEGX210
MIEGX220
MIEGX230
MIEGX240
MIEGX250
MIEGX260
MIEGX270
MIEGX280
MIEGX290
MIEGX300
MIEGX310
MIEGX320
MIEGX330
MIEGX340
MIEGX350
MIEGX360
MIEGX370
MIEGX380
MIEGX390
MIEGX400
MIEGX410
MIEGX420
MIEGX430
MIEGX440
MIEGX450
MIEGX460
MIEGX470
MIEGX480
MIEGX490
MIEGX500
MIEGX510
MIEGX520
MIEGX530
MIEGX540
MIEGX550
MIEGX560
MIEGX570
MIEGX580
MIEGX590
MIEGX600
MIEGX610
MIEGX620
MIEGX630
MIEGX640
MIEGX650
MIEGX660
MIEGX670
MIEGX680
MIEGX690
MIEGX700

```

```

NMIN=NMX+1-NDELTA
V=FLOAT(NMX)+0.50
Y=RF*X
ZANP=2.0/Y
ZNUM=ZANP*Y
ZPDT=ZNUM
V=V+1.0
ZDEN=ZANP*V
ZNUM=ZDEN-1.0/ZNUM
5 ZRPDT=ZNUM/ZDEN
ZPDT=ZRPDT*ZPDT
IF (ABS(REAL(ZRPDT)-1.0),LT. TOL) GO TO 7
IF (V.LT.20000.0) GO TO 6
WRITE (IOUT,1000) X,EM,CAYE
STOP
6 V=V+1.0
ZAN=ZANP*V
ZNUM=ZAN-1.0/ZNUM
ZDEN=ZAN-1.0/ZDEN
GO TO 5
7 CONTINUE
J=NMX
8 JJ=J-NMX+NDELTA
A(JJ)=-(FLOAT(J))/Y+ZPDT
J=J-1
IF (J.LT.NMIN) GO TO 9
ZPDT=(2.0*FLOAT(J)+1.0)/Y-1.0/ZPDT
GO TO 8
9 CONTINUE
J=N-NMX+NDELTA
ACAPN=A(J)
IF (N.GT.1) GO TO 11
C THIS PART FOR N EQUAL 1 ONLY
TC1=ACAPN*RRF+RX
TC2=ACAPN*RF+RX
C SEE EQUIVALENCE STMTS FOR EXPLANATION OF TAC ), ETC.
FNA=(TC1*TAC(3)-TAC(1))/(TC1*WFN(2)-WFN(1))
FNB=(TC2*TAC(3)-TAC(1))/(TC2*WFN(2)-WFN(1))
FNAP=FNA
FNBP=FNB
T(1)=1.50
TB(1)=T(1)*TB(1)
TB(2)=T(1)*TB(2)
TC(1)=T(1)*TC(1)
TC(2)=T(1)*TC(2)
DO 10 J=1,IT
TAU2J=TAU(2,J)
ELTRMX(1,J)=TB(1)+TC(1)*TAU2J
ELTRMX(2,J)=TB(2)+TC(2)*TAU2J
ELTRMX(3,J)=TC(1)+TB(1)*TAU2J
ELTRMX(4,J)=TC(2)+TB(2)*TAU2J
10 CONTINUE
QEXT=2.0*(TB(1)+TC(1))
QSCAT=(TB(1)*TB(1)+TB(2)*TB(2)+TC(1)*TC(1)+TC(2)*TC(2))/0.750
Q1STAR=0.0
Q2STAR=0.0
SUMRR=2.0*(TB(1)-TC(1))
SUMRI=2.0*(TB(2)-TC(2))
N=2
GO TO 2
11 CONTINUE
TC1=ACAPN*RRF+EN*RX
TC2=ACAPN*RF+EN*RX
C SEE EQUIVALENCE STMTS FOR EXPLANATION OF TAC ), ETC.
FNA=(TC1*TAC(3)-TAC(1))/(TC1*WFN(2)-WFN(1))
FNB=(TC2*TAC(3)-TAC(1))/(TC2*WFN(2)-WFN(1))
T(4)=T(1)/(EN*T(2))
T(2)=(T(2)*(EN+1.0))/EN
S=-S
SUMRR=SUMRR+S*T(3)*(TB(1)-TC(1))

```

```

MIEGX710
MIEGX720
MIEGX730
MIEGX740
MIEGX750
MIEGX760
MIEGX770
MIEGX780
MIEGX790
MIEGX800
MIEGX810
MIEGX820
MIEGX830
MIEGX840
MIEGX850
MIEGX860
MIEGX870
MIEGX880
MIEGX890
MIEGX900
MIEGX910
MIEGX920
MIEGX930
MIEGX940
MIEGX950
MIEGX960
MIEGX970
MIEGX980
MIEGX990
MIEGX000
MIEGX010
MIEGX020
MIEGX030
MIEGX040
MIEGX050
MIEGX060
MIEGX070
MIEGX080
MIEGX090
MIEGX100
MIEGX110
MIEGX120
MIEGX130
MIEGX140
MIEGX150
MIEGX160
MIEGX170
MIEGX180
MIEGX190
MIEGX200
MIEGX210
MIEGX220
MIEGX230
MIEGX240
MIEGX250
MIEGX260
MIEGX270
MIEGX280
MIEGX290
MIEGX300
MIEGX310
MIEGX320
MIEGX330
MIEGX340
MIEGX350
MIEGX360
MIEGX370
MIEGX380
MIEGX390
MIEGX400

```

```

C      SUMRI=SUMRI+S*T(3)*(TB(2)-TC(2))
      SEE LATER COMMENTS ABOUT FOLLOWING STATEMENTS
      QRTL1=QRT
      QRT=SUMRR*SUMRR+SUMRI*SUMRI
C      01STAR CALCULATION
      01STAR=01STAR+(TB(1)*TD(1)+TB(2)*TD(2)+TC(1)*TE(1)+TC(2)*TE(2))
1      *T(2)*4.0+4.0*T(4)*(TD(1)*TE(1)+TD(2)*TE(2))
C      IF (N.LT.3) GO TO 12
      02STAR CALCULATION
      F1=TF(1)*TF(1)+TF(2)*TF(2)+TG(1)*TG(1)+TG(2)*TG(2)
      F2=TB(1)*TF(1)+TB(2)*TF(2)+TC(1)*TG(1)+TC(2)*TG(2)
      F3=TD(1)*TG(1)+TD(2)*TG(2)+TE(1)*TF(1)+TE(2)*TF(2)
      ENL1=EN-1.0
      COF1=2.50*((EN-2.0)*ENL1-3.0)*((EN-2.0)*ENL1-3.0)*(2.0*EN-3.0)/
1      ((EN-2.0)*ENL1*(2.0*EN-1.0)*(2.0*EN-5.0))
      COF2=7.50*(EN-2.0)*(EN+1.0)/(2.0*EN-1.0)
      COF3=15.0/ENL1
      02STAR=02STAR+COF1*F1+COF2*F2+COF3*F3
12     CONTINUE
      QEXT=QEXT+T(3)*(TB(1)+TC(1))
      T(4)=TB(1)*TB(1)+TB(2)*TB(2)+TC(1)*TC(1)+TC(2)*TC(2)
      QSCAT=QSCAT+T(3)*T(4)
      T(2)=EN*(EN+1.0)
      T(1)=T(3)/T(2)
      DO 13 J=1,IT
      PI3J=PI(3,J)
      TAU3J=TAU(3,J)
      ELTRMX(1,J)=ELTRMX(1,J)+T(1)*(TB(1)*PI3J+TC(1)*TAU3J)
      ELTRMX(2,J)=ELTRMX(2,J)+T(1)*(TB(2)*PI3J+TC(2)*TAU3J)
      ELTRMX(3,J)=ELTRMX(3,J)+T(1)*(TC(1)*PI3J+TB(1)*TAU3J)
      ELTRMX(4,J)=ELTRMX(4,J)+T(1)*(TC(2)*PI3J+TB(2)*TAU3J)
13     CONTINUE
      IF (N.LT.5) GO TO 14
      QRTR=ABS((QRT-QRTL1)/QRT)
      TEST FOR CONVERGENCE ON QEXT, QSCA, AND QRADAR
      IF ((T(4).LT. TOL ).AND.(QRTR.LT. TOL )) GO TO 16
14     N=N+1
      DO 15 J=1,IT
      PI(1,J)=PI(2,J)
      PI(2,J)=PI(3,J)
      TAU(1,J)=TAU(2,J)
      TAU(2,J)=TAU(3,J)
15     CONTINUE
      FNAPP=FNAP
      FNBPP=FNBP
      FNAP=FNA
      FNBP=FNB
      GO TO 2
16     CONTINUE
      DO 18 J=1,IT
      DO 17 I=1,4
      T(I)=ELTRMX(I,J)
17     CONTINUE
      ELTRMX(1,J)=T(3)*T(3)+T(4)*T(4)
      ELTRMX(2,J)=T(1)*T(1)+T(2)*T(2)
      ELTRMX(3,J)=T(1)*T(3)+T(2)*T(4)
      ELTRMX(4,J)=(T(2)*T(3)-T(4)*T(1))
      PFNZRO=(ELTRMX(1,J)+ELTRMX(2,J))/2.0
      IF (IAPXCT.EQ.1) GO TO 20
      P(J)=PFNZRO
18     CONTINUE
      ELTRMX(2,J) IS THE VERTICAL COMPONENT SCATTERING I1 (EYE1)
      ELTRMX(1,J) IS THE HORIZONTAL COMPONENT SCATTERING I2 (EYE2)
      ELTRMX(3,J) IS EQUIVALENT TO EYE3
      ELTRMX(4,J) IS EQUIVALENT TO -1.0*EYE4
      T(1)=2.0*RX*RX
      SGT=QEXT*T(1)
      SGS=QSCAT*T(1)
      01STAR=3.0*01STAR/(X*X*SGT)
      02STAR=4.0*02STAR/(X*X*SGT)

```

MIEGX410
 MIEGX420
 MIEGX430
 MIEGX440
 MIEGX450
 MIEGX460
 MIEGX470
 MIEGX480
 MIEGX490
 MIEGX500
 MIEGX510
 MIEGX520
 MIEGX530
 MIEGX540
 MIEGX550
 MIEGX560
 MIEGX570
 MIEGX580
 MIEGX590
 MIEGX600
 MIEGX610
 MIEGX620
 MIEGX630
 MIEGX640
 MIEGX650
 MIEGX660
 MIEGX670
 MIEGX680
 MIEGX690
 MIEGX700
 MIEGX710
 MIEGX720
 MIEGX730
 MIEGX740
 MIEGX750
 MIEGX760
 MIEGX770
 MIEGX780
 MIEGX790
 MIEGX800
 MIEGX810
 MIEGX820
 MIEGX830
 MIEGX840
 MIEGX850
 MIEGX860
 MIEGX870
 MIEGX880
 MIEGX890
 MIEGX900
 MIEGX910
 MIEGX920
 MIEGX930
 MIEGX940
 MIEGX950
 MIEGX960
 MIEGX970
 MIEGX980
 MIEGX990
 MIEGX000
 MIEGX010
 MIEGX020
 MIEGX030
 MIEGX040
 MIEGX050
 MIEGX060
 MIEGX070
 MIEGX080
 MIEGX090
 MIEGX100

	SGR=(SUMRR*SUMRR+SUMRI*SUMRI)*RX*RX	MIEGX110
	QTD=(SGT)	MIEGX120
	QSD=(SGS)	MIEGX130
	QRD=(SGR)	MIEGX140
	O1STRD=(O1STAR)	MIEGX150
	O2STRD=(O2STAR)	MIEGX160
C	LOOP FOR CALCULATION OF PFN AT ZERO DEGREES - FOR GPHASX	MIEGX170
	IAPXCT=1	MIEGX180
	IT=1	MIEGX190
	GO TO 21	MIEGX200
20	IT=ITT	MIEGX210
	RETURN	MIEGX220
1000	FORMAT (52H V GT 20000 ERROR IN CONTINUED FRACTIONS MIE ROUTINE,	MIEGX230
1	11H ** ALPHA =,E12.6,6H EM = ,E12.6,7H CAY = ,E12.6,	MIEGX240
2	1X,54H IT IS SUGGESTED THAT TOL=1.E-06 FOR SINGLE PRECISION.)	MIEGX250
	END	MIEGX260

```

SUBROUTINE WATER(WAVE, EMT, CAYT, TMCHUR, RHODEN)
COMMON /IO/ IOIN, IOOUT, NUNIT, IEO, NEOU
REAL LAMBDA(169), NSUBR(169), NSUBI(169)
DIMENSION TEMP(7), DENS(7)
C THIS PROGRAM SEARCHES HALE AND QUERRY TABLE FOR REFRACTION VS.
C WAVELENGTH (APPLIED OPTICS, VOL. 12, NO. 3, MARCH 1973, PG 555)
C AND THE DENSITY VS. TEMPERATURE (HANDBK OF CHEM AND PHYS).
C IF THE INPUT VALUES (TMCHUR AND WAVE) ARE NOT IN THE TABLES, A
C LINEAR INTERPOLATION IS COMPUTED. VALUES ACCURATE TO THREE PLACES.
INTEGER P, POINT, H
C
C TABLES FOLLOW FOR 60 LINES
DATA LAMBDA/0.200,0.225,0.250,0.275,0.300,0.325,0.350,0.375,0.400
1, 0.425,0.450,0.475,0.500,0.525,0.550,0.575,0.600,0.625,0.650
2, 0.675,0.700,0.725,0.750,0.775,0.800,0.825,0.850,0.875,0.900
3, 0.925,0.950,0.975,1.000,1.200,1.400,1.600,1.800,2.000,2.200
4, 2.400,2.600,2.800,3.000,3.200,3.400,3.600,3.800,4.000,4.200
5, 3.050,3.100,3.150,3.200,3.250,3.300,3.350,3.400,3.450,3.500
6, 3.600,3.700,3.800,3.900,4.000,4.100,4.200,4.300,4.400,4.500
7, 4.600,4.700,4.800,4.900,5.000,5.100,5.200,5.300,5.400,5.500
8, 5.600,5.700,5.800,5.900,6.000,6.100,6.200,6.300,6.400,6.500
9, 6.600,6.700,6.800,6.900,7.000,7.100,7.200,7.300,7.400,7.500
X, 7.600,7.700,7.800,7.900,8.000,8.200,8.400,8.600,8.800,9.000
1, 9.200,9.400,9.600,9.800,10.00,10.50,11.00,11.50,12.00,12.50
2, 13.00,13.50,14.00,14.50,15.00,15.50,16.00,16.50,17.00,17.50
3, 18.00,18.50,19.00,19.50,20.00,21.00,22.00,23.00,24.00,25.00
4, 26.00,27.00,28.00,29.00,30.00,32.00,34.00,36.00,38.00,40.00
5, 42.00,44.00,46.00,48.00,50.00,60.00,70.00,80.00,90.00,100.0
6, 110.0,120.0,130.0,140.0,150.0,160.0,170.0,180.0,190.0,200.0
7,
DATA NSUBR/1.396,1.373,1.362,1.354,1.349,1.346,1.343,1.341,1.339
1, 1.338,1.337,1.336,1.335,1.334,1.333,1.333,1.332,1.332,1.331
2, 1.331,1.331,1.330,1.330,1.330,1.329,1.329,1.329,1.328,1.328
3, 1.328,1.327,1.327,1.327,1.324,1.321,1.317,1.312,1.306,1.296
4, 1.279,1.242,1.219,1.188,1.157,1.142,1.149,1.201,1.292,1.371
5, 1.426,1.467,1.483,1.478,1.467,1.450,1.432,1.420,1.410,1.400
6, 1.385,1.374,1.364,1.357,1.351,1.346,1.342,1.338,1.334,1.332
7, 1.330,1.330,1.330,1.328,1.325,1.322,1.317,1.312,1.305,1.298
8, 1.289,1.277,1.262,1.248,1.265,1.319,1.363,1.357,1.347,1.339
9, 1.334,1.329,1.324,1.321,1.317,1.314,1.312,1.309,1.307,1.304
X, 1.302,1.299,1.297,1.294,1.291,1.286,1.281,1.275,1.269,1.262
1, 1.255,1.247,1.239,1.229,1.218,1.185,1.153,1.126,1.111,1.123
2, 1.146,1.177,1.210,1.241,1.270,1.297,1.325,1.351,1.376,1.401
3, 1.423,1.443,1.461,1.476,1.480,1.487,1.500,1.511,1.521,1.531
4, 1.539,1.545,1.549,1.551,1.551,1.546,1.536,1.527,1.522,1.519
5, 1.522,1.530,1.541,1.555,1.587,1.703,1.821,1.886,1.924,1.957
6, 1.966,2.004,2.036,2.056,2.069,2.081,2.094,2.107,2.119,2.130
7,
DATA NSUBI/1.10E-7,4.90E-8,3.35E-8,2.35E-8,1.60E-8,1.08E-8,6.50E-9
1, 3.50E-9,1.86E-9,1.30E-9,1.02E-9,9.35E-10,1.00E-9,1.32E-9,1.96E-9
2, 3.60E-9,1.09E-8,1.39E-8,1.64E-8,2.23E-8,3.35E-8,9.15E-8,1.56E-7
3, 1.48E-7,1.25E-7,1.82E-7,2.93E-7,3.91E-7,4.86E-7,1.06E-6,2.93E-6
4, 3.48E-6,2.89E-6,9.89E-6,1.38E-5,8.55E-5,1.15E-4,1.10E-3,2.89E-4
5, 9.56E-4,3.17E-3,6.70E-3,1.90E-2,5.90E-2,1.15E-1,1.85E-1,2.68E-1
6, 2.98E-1,2.72E-1,2.40E-1,1.92E-1,1.35E-1,9.24E-2,6.10E-2,3.68E-2
7, 2.61E-2,1.95E-2,1.32E-2,9.40E-3,5.15E-3,3.60E-3,3.40E-3,3.80E-3
8, 4.60E-3,3.52E-3,6.88E-3,8.45E-3,1.03E-2,1.34E-2,1.47E-2,1.57E-2
9, 1.50E-2,1.37E-2,1.24E-2,1.11E-2,1.01E-2,9.80E-3,1.03E-2,1.16E-2
X, 1.42E-2,2.03E-2,3.30E-2,6.22E-2,1.07E-1,1.31E-1,8.80E-2,5.70E-2
1, 4.49E-2,2.392E-2,3.56E-2,3.37E-2,3.27E-2,3.22E-2,3.20E-2,3.20E-2
2, 3.21E-2,3.22E-2,3.24E-2,3.26E-2,3.28E-2,3.31E-2,3.35E-2,3.39E-2
3, 3.43E-2,3.31E-2,3.61E-2,3.72E-2,3.85E-2,3.99E-2,4.15E-2,4.33E-2
4, 4.54E-2,4.49E-2,4.79E-2,5.08E-2,5.62E-2,6.68E-2,1.42E-1,1.99E-1
5, 3.05E-1,3.43E-1,3.70E-1,3.88E-1,4.02E-1,4.14E-1,4.22E-1,4.28E-1
6, 4.29E-1,4.29E-1,4.29E-1,4.21E-1,4.14E-1,4.04E-1,3.93E-1,3.82E-1
7, 3.73E-1,3.67E-1,3.61E-1,3.56E-1,3.50E-1,3.44E-1,3.38E-1,3.33E-1
8, 3.28E-1,3.24E-1,3.29E-1,3.43E-1,3.61E-1,3.85E-1,4.09E-1,4.36E-1
9, 4.62E-1,4.88E-1,5.14E-1,5.87E-1,5.76E-1,5.47E-1,5.36E-1,5.32E-1
X, 5.31E-1,5.26E-1,5.14E-1,5.00E-1,4.95E-1,4.96E-1,4.97E-1,4.99E-1
1, 5.01E-1,5.04E-1,
C ALTERNATE FORM OF ABOVE DATA STMT DUE TO EXCESS CONTINUATION CARDS

```

```

DATA (NSUBI(I), I=1,143) WAT00710
+ 1.10E-7, 4.90E-8, 3.35E-8, 2.35E-8, 1.60E-8, 1.08E-8, 6.50E-9 WAT00720
1, 3.50E-9, 1.86E-9, 1.30E-9, 1.02E-9, 9.35E-10, 1.00E-9, 1.32E-9, 1.96E-9 WAT00730
2, 3.60E-9, 1.09E-8, 1.39E-8, 1.64E-8, 2.23E-8, 3.35E-8, 9.15E-8, 1.56E-7 WAT00740
3, 1.48E-7, 1.25E-7, 1.82E-7, 2.93E-7, 3.91E-7, 4.86E-7, 1.06E-6, 2.93E-6 WAT00750
4, 3.48E-6, 2.89E-6, 9.89E-6, 1.38E-4, 8.55E-5, 1.15E-4, 1.10E-3, 2.89E-4 WAT00760
5, 9.56E-4, 3.17E-3, 6.70E-3, 1.90E-2, 5.90E-2, 1.15E-1, 1.85E-1, 2.68E-1 WAT00770
6, 2.98E-1, 2.72E-1, 2.40E-1, 1.92E-1, 1.35E-1, 9.24E-2, 6.10E-2, 3.68E-2 WAT00780
7, 2.61E-2, 1.95E-2, 1.32E-2, 9.40E-3, 5.15E-3, 3.60E-3, 3.40E-3, 3.80E-3 WAT00790
8, 4.60E-3, 5.62E-3, 6.88E-3, 8.45E-3, 1.03E-2, 1.34E-2, 1.47E-2, 1.57E-2 WAT00800
9, 1.50E-2, 1.37E-2, 1.24E-2, 1.11E-2, 1.01E-2, 9.80E-3, 1.03E-2, 1.16E-2 WAT00810
X, 1.42E-2, 2.03E-2, 3.30E-2, 6.22E-2, 1.07E-1, 1.31E-1, 1.8.80E-1, 5.70E-1 WAT00820
1, 4.49E-2, 3.92E-2, 3.56E-2, 3.37E-2, 3.27E-2, 3.22E-2, 3.20E-2, 3.20E-2 WAT00830
2, 3.21E-2, 3.22E-2, 3.24E-2, 3.26E-2, 3.28E-2, 3.31E-2, 3.35E-2, 3.39E-2 WAT00840
3, 3.43E-2, 3.51E-2, 3.61E-2, 3.72E-2, 3.85E-2, 3.99E-2, 4.15E-2, 4.33E-2 WAT00850
4, 4.54E-2, 4.79E-2, 5.08E-2, 5.62E-2, 6.68E-2, 9.14E-2, 1.19E-1, 1.59E-1 WAT00860
5, 3.05E-1, 3.43E-1, 3.70E-1, 3.88E-1, 4.02E-1, 4.14E-1, 4.22E-1, 4.28E-1 WAT00870
6, 4.29E-1, 4.29E-1, 4.26E-1, 4.21E-1, 4.14E-1, 4.04E-1, 3.93E-1, 3.82E-1 WAT00880
7, 3.73E-1, 3.67E-1, 3.61E-1, 3.56E-1, 3.50E-1, 3.44E-1, 3.38E-1, 3.33E-1 WAT00890
DATA (NSUBI(I), I=144,169) / WAT00900
8, 3.28E-1, 3.24E-1, 3.29E-1, 3.43E-1, 3.61E-1, 3.85E-1, 4.09E-1, 4.36E-1 WAT00910
9, 4.62E-1, 4.88E-1, 5.14E-1, 5.87E-1, 5.76E-1, 5.47E-1, 5.36E-1, 5.32E-1 WAT00920
X, 5.31E-1, 5.26E-1, 5.14E-1, 5.00E-1, 4.95E-1, 4.96E-1, 4.97E-1, 4.99E-1 WAT00930
1, 5.01E-1, 5.04E-1 / WAT00940
DATA TEMP/273., 278., 283., 288., 293., 298., 303. / DENS/ WAT00950
1, 0.999841, 0.999965, 0.999700, 0.999099, 0.998203, 0.997044, WAT00960
2, 0.995994 / WAT00970
EMT=0.0 WAT00980
CAYT=0.0 WAT00990
POINT=0 WAT01000
H=0 WAT01010
IF (WAVE.LT.0.2.OR.WAVE.GT.200.0.OR.TMCHUR.LT.273.0R. WAT01020
1, TMCHUR.GT.303.0) GO TO 11 WAT01030
C BINARY SEARCH WAT01040
L=1 WAT01050
H=125 WAT01060
1 POINT=((L+H)/2) WAT01070
TEST=ABS(LAMBDA(POINT)-WAVE) WAT01080
IF (TEST.LE.0.0001) GO TO 4 WAT01090
IF (WAVE.GT.LAMBDA(POINT)) GO TO 2 WAT01100
H=POINT WAT01110
GO TO 3 WAT01120
2 L=POINT+1 WAT01130
3 IF (L.NE.H) GO TO 1 WAT01140
L=L-1 WAT01150
C INTERPOLATION ROUTINE WAT01160
EMT=NSUBR(L)+(NSUBR(L+1)-NSUBR(L))*((WAVE-LAMBDA(L))/(LAMBDA(L+1)- WAT01170
1, LAMBDA(L))) WAT01180
CAYT=NSUBI(L)+(NSUBI(L+1)-NSUBI(L))*((WAVE-LAMBDA(L))/(LAMBDA(L+1)- WAT01190
1, -LAMBDA(L))) WAT01200
GO TO 5 WAT01210
4 CONTINUE WAT01220
EMT=NSUBR(POINT) WAT01230
CAYT=NSUBI(POINT) WAT01240
C SEARCH TEMP VS DENS WAT01250
5 IF (TMCHUR.LT.273.0.OR.TMCHUR.GT.303.0) GO TO 11 WAT01260
L=1 WAT01270
H=7 WAT01280
6 P=((L+H)/2) WAT01290
TESTT=ABS(TEMP(P)-TMCHUR) WAT01300
IF (TESTT.LE.0.0001) GO TO 9 WAT01310
IF (TMCHUR.GT.TEMP(P)) GO TO 7 WAT01320
H=P WAT01330
GO TO 8 WAT01340
7 L=P+1 WAT01350
8 IF (L.NE.H) GO TO 6 WAT01360
L=L-1 WAT01370
RHODEN=DENS(L)+(DENS(L+1)-DENS(L))*((TMCHUR-TEMP(L))/(TEMP(L+1)- WAT01380
1, TEMP(L))) WAT01390
GO TO 10 WAT01400

```

9 RHODEN=DENS(P)	UAT 01410
10 CONTINUE	UAT 01420
GO TO 13	UAT 01430
11 WRITE (IOUT,12) TMCHUR,WAVE	UAT 01440
12 FORMAT (10H TEMP. OF .E12.6,14H OR WAVE. OF .E12.6,14H BEYOND RAN	UAT 01450
1GE,24HOF DATA IN WATER ROUTINE/22H EXECUTION TERMINATED)	UAT 01460
STOP	UAT 01470
13 RETURN	UAT 01480
END	UAT 01490

	SUBROUTINE GAUS(IT)	GAUS0010
	COMMON /PT2/ PC(65),OL(65),RMS(65),PSUM(65),PSUMT(65),P(65)	GAUS0020
	COMMON /IO/ IOIN,IOOUT,NUNIT,IEO,NEOU	GAUS0030
	COMMON /AGXM/ C(65),W(65),OLT(65),JDIMCK(3)	GAUS0040
	WRITE (IOOUT,11)	GAUS0050
C	INITIALIZE ARRAY PC() USED FOR RUNNING SUMMATION	GAUS0060
	DO 1 I=1,IT	GAUS0070
	OL(I)=0.	GAUS0080
1	PC(I)=0.E+00	GAUS0090
C	LOOPS 2 AND 3 CALCULATE EXPANSION COEFS. FOR FUNCTION PSUM()	GAUS0100
C	VIA GAUSS-LEGENDRE QUADRATURE. THE COEFS. GO INTO ARRAY OL()	GAUS0110
	DO 3 I=1,IT	GAUS0120
	COF=W(I)*PSUM(I)	GAUS0130
	PLM1=C(I)	GAUS0140
	PLM2=1.	GAUS0150
	DO 2 LL=1,IT	GAUS0160
	L=LL-1	GAUS0170
	PL=PLM2	GAUS0180
	IF (LL.EQ.2) PL=PLM1	GAUS0190
	IF (LL.LE.2) GO TO 2	GAUS0200
	PL=2.*C(I)*PLM1-PLM2-(C(I)*PLM1-PLM2)/FLOAT(L)	GAUS0210
	PLM2=PLM1	GAUS0220
	PLM1=PL	GAUS0230
2	OL(LL)=OL(LL)+COF*PL*(FLOAT(L)+.5)	GAUS0240
3	CONTINUE	GAUS0250
	DO 7 I=1,IT	GAUS0260
	II=I-1	GAUS0270
7	WRITE (IOOUT,8) II,OL(I)	GAUS0280
8	FORMAT (1H ,20X,I6,20X,1PE14.7)	GAUS0290
11	FORMAT (1H /,1X,25X,1HL,20X,16HL-TN COEFFICIENT)	GAUS0300
	RETURN	GAUS0310
	END	GAUS0320

	SUBROUTINE DIMER(NGO)	DIM00010
	COMMON /IO/ IOIN, IOUT, NUNIT, IEQ, NEQU	DIM00020
	GO TO (1,2,3), NGO	DIM00030
1	WRITE(IOUT,101)	DIM00040
101	FORMAT(1H, '**** THE INPUT VALUE OF IT IS GREATER '	DIM00050
	1, 'THAN THE ARRAY DIMENSIONS', //, 'CHANGE THE DIMENSIONS OF THE '	DIM00060
	2, 'FOLLOWING ARRAYS IN SUBS AND COMMON', //, 'COMMON BLOCK', 10X,	DIM00070
	3, 'ARRAY(S)', //, 'AGXM', 10X, 'C, H, OLT', //, 'PT2', 11X,	DIM00080
	4, 'PC, OL, RMS, PSUM, PSUMT, P', //, 'ALSO CHANGE THE VALUE OF JIMCK(1) '	DIM00090
	5, 'IN THE DATA STMT TO AGREE WITH THE NEW', //, 'DIMENSION LIMITS',	DIM00100
	6, 1H, 55H 'THE SECOND INDEX ON ARRAYS PI, TAU, AND ELTRMX IN MIEGX	DIM00110
	7, 1H, 56H 'MUST ALSO BE CHANGED AND ARRAYS P AND C CHANGED AS WELL. '	DIM00120
	STOP	DIM00130
2	WRITE (IOUT,102)	DIM00140
102	FORMAT(1H, '**** TOO MANY PARTICLE RADII FOR DIMENSION LIMITS: '	DIM00150
	1, 'IN SUBS AND COMMON CHANGE THE FOLLOWING '	DIM00160
	2, 'ARRAYS', //, 'COMMON BLOCK', 10X, 'ARRAY(S)', //, 'PT1', 11X, 'F, R, FF',	DIM00170
	3, 'ARRAYS F AND R MUST BE CONSISTENT WITH THE FOLLOWING: ARRAY '	DIM00180
	4, 'SIZE = 1 + 2**JDIMCK(2)', //, 'ARRAY FF MUST BE DIMENSIONED TO ONE',	DIM00190
	5, 'MORE THAN ARRAYS F AND R', //, 'ALSO CHANGE THE VALUE OF JDIMCK(2)',	DIM00200
	6, 'IN THE DATA STATEMENT')	DIM00210
	STOP	DIM00220
3	WRITE (IOUT,103)	DIM00230
103	FORMAT (1H, ' THE DIMENSIONS OF F AND '	DIM00240
	1, 'R DO NOT AGREE WITH THE FOLLOWING: ', //, 'SIZE=1+2**JDIMCK(2) ',	DIM00250
	2, 'WHERE JDIMCK(2) APPEARS IN THE DATA STATEMENT', //, 'ALSO DIMENSION	DIM00260
	3, 'ARRAY FF TO BE ONE MORE THAN ARRAYS F AND R')	DIM00270
	STOP	DIM00280
	END	DIM00290

```

C      BLOCK DATA
C      IF YOU CHANGE THE DIMENSIONS MAKE SURE THAT YOU ALSO CHANGE THE
C      DATA STATEMENT CONTAINING JDIMCK( )
C      JDIMCK(1)=ORDER OF QUADRATURE (INPUT IT), I.E. MAX SIZE OF DIMENSION
C      JDIMCK(2):USED IN COMPUTATION OF SIZE OF ARRAYS F AND R,
C      SIZE=1+2**JDIMCK(2), ARRAY FF SHOULD BE ONE MORE THAN F AND R.
C      JIMCK(3) WILL BE CALCULATED.
C      THE OTHER ARRAYS ARE REUSED, SO
C      CHECK THE SUBROUTINE CALLS AND COMMON BLOCKS TO SEE IF
C      ARRAYS HAVE BEEN RENAMED WHEN U CHANGE DIMENSIONS.
COMMON /AQXM/ C(65),W(65),OLT(65),JDIMCK(3)
COMMON /PT1/ F(513),R(513),DR(8),RR(9),FF(514)
+ ,NRADI,PI,IDSTP,NKG,NHALV,NI
COMMON /PT2/ PC(65),OL(65),RMS(65),PSUM(65),PSUMT(65),P(65)
COMMON /IO/ IOIN,IOUT,NUNIT,IEO,NEOU
DATA IOIN,IOUT,NUNIT,IEO,NEOU /5,6,3*0/
C      DATA JDIMCK /65,9,0/
EOSAEL OPTION: 65 PREDETERMINED ANGLES
DATA W
+ 0.0, 0.5, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 8.0, 10.0,
+ 12.0, 14.0, 16.0, 18.0, 20.0, 24.0, 28.0, 32.0, 36.0, 40.0,
+ 44.0, 48.0, 52.0, 56.0, 60.0, 64.0, 68.0, 72.0, 76.0, 80.0,
+ 84.0, 88.0, 92.0, 96.0, 100.0, 104.0, 108.0, 112.0, 116.0, 120.0,
+ 124.0, 128.0, 132.0, 136.0, 140.0, 142.0, 144.0, 146.0, 148.0, 150.0,
+ 152.0, 154.0, 156.0, 158.0, 160.0, 162.0, 164.0, 166.0, 168.0, 170.0,
+ 172.0, 174.0, 176.0, 178.0, 180.0/
END

```

```

BLK00010
BLK00020
BLK0003
BLK00040
BLK00050
BLK00060
BLK00070
BLK00080
BLK00090
BLK00100
BLK00110
BLK00120
BLK00130
BLK00140
BLK00150
BLK00160
BLK00170
BLK00180
BLK00190
BLK00200
BLK00210
BLK00220
BLK00230
BLK00240
BLK00250
BLK00260
BLK00270

```

PROGRAM FLASH1	DRIV0010
COMMON/IOUNT/IOIN, IOOUT	DRIV0020
COMMON/CONST/PI	DRIV0030
COMMON/IDATA/ROBS(3), RTAR(3), TARDEG, RSRG(3), TIMGUN, FOV, WAVE,	DRIV0040
* RADSEE, RADLOC, TARLEN, TARWID, TARHGT, PCTSEE, PCTLOC,	DRIV0050
* TIME, ISRC, RCTSEE, RCTLOC	DRIV0060
PI=3.14159	DRIV0070
IOIN=5	DRIV0080
IOOUT=6	DRIV0090
C**** NOTE ICHK=-1 DEFAULTS USETR INPUT TO WAVE1	DRIV0100
ICLK=0	DRIV0110
WAVE1=.55	DRIV0120
CALL FLASH(WAVE1, ICHK)	DRIV0130
WRITE (IOOUT, 100) WAVE1, ICHK	DRIV0140
100 FORMAT (5X, 13HEOSAEI OUTPUT, /,	DRIV0150
*5X, 13HWAVELENGTH = , F6.1, 1X, 11HMICROMETERS, /,	DRIV0160
*5X, 7HICK = , I4)	DRIV0170
STOP	DRIV0180
END	

```

SUBROUTINE FLASH(WAVE1,ICLK)
C*****
C*          PROGRAM FLASH
C*          EOSAEL80
C*****
DIMENSION FLASH1(3),FLASH2(3),IC1(3),IC2(3),IGUN(3)
COMMON/IOUNT/IOIN,IOOUT
COMMON/CONST/PI
COMMON/IDATA/ROBS(3),RTAR(3),TARDEG,RSRC(3),TIMGUN,F0V,WAVE,
*          RADSEE,RADLOC,TARLEN,TARWID,TARHGT,PCTSEE,PCTLOC,
*          TIME,ISRC,RCISEE,RCITOC
DATA FLASH1/3.85,2.55,0.0/
DATA FLASH2/0.15,0.15,0.0/
DATA IC1/2HMO,2HT-,2HUN/
DATA IC2/2HW,,2H55,2HKN/
DATA IGUN/105,100,0/
98 FORMAT(//,21X,30H*****PROGRAM FLASH OUTPUT*****,/,21X,40(2H--))
99 FORMAT(21X,20H*****END OF RUN*****,/,21X,40(2H--))
100 FORMAT(1H1,20X,40(2H**),/,21X,1H*,34X,13HPROGRAM FLASH,31X,1H*,/,
*21X,1H*,37X,8HEOSAEL80,33X,1H*,/,21X,40(2H**))
101 FORMAT(//,21X,15H*****INPUT*****,/,21X,40(2H--))
102 FORMAT(21X,14HSCENARIO DATA:,,21X,14HREFERENCE TIME,1X,F8.3,
*1X,3HSEC)
103 FORMAT(21X,9HOBSERVER:,18X,7HTARGET:,20X,7HSOURCE:)
104 FORMAT(24X,6HX(OBS),3X,F6.1,2H M,10X,6HX(TAR),3X,F6.1,2H M,10X,
*6HX(SRC),3X,F6.1,2H M,/,
*24X,6HY(OBS),3X,F6.1,2H M,10X,6HY(TAR),3X,F6.1,2H M,10X,
*6HY(SRC),3X,F6.1,2H M,/,
*24X,6HZ(OBS),3X,F6.1,2H M,10X,6HZ(TAR),3X,F6.1,2H M,10X,
*6HZ(SRC),3X,F6.1,2H M)
105 FORMAT(48X,11HORIENTATION,1X,F6.1,1X,3HDEG,8X,
*10HEVENT TIME,1X,F6.3,1X,3HSEC,/,67X,
*10HCCW X-AXIS)
106 FORMAT(21X,25HDETECTOR CHARACTERISTICS:,,
*24X,13HFIELD OF VIEW,12X,F6.1,1X,7HDEGREES,/,
*24X,10HWAVELENGTH,15X,F6.1,1X,11HMICROMETERS,/,
*24X,21HRESOLUTION CRITERIA--/,
*26X,17H(A) FOR DETECTION,6X,F6.3,1X,12HMILLIRADIANS,/,
*26X,15H(B) FOR LOCK ON,8X,F6.3,1X,12HMILLIRADIANS,/,
*24X,24HRECOVERY TIME (R=100M)--/,
*26X,17H(A) FOR DETECTION,6X,F6.1,1X,7HSECONDS,/,
*26X,15H(B) FOR LOCK ON,8X,F6.1,1X,7HSECONDS)
107 FORMAT(21X,23HTARGET CHARACTERISTICS:,,
*24X,6HLENGTH,19X,F6.1,1X,6HMETERS,/,
*24X,5HWIDTH,20X,F6.1,1X,6HMETERS,/,
*24X,6HEIGHT,19X,F6.1,1X,6HMETERS,/,
*24X,19HEXPOSURE CRITERIA--/,
*26X,17H(A) FOR DETECTION,6X,F6.1,1X,7HPERCENT,/,
*26X,15H(B) FOR LOCK ON,8X,F6.1,1X,7HPERCENT)
108 FORMAT(21X,23HSOURCE CHARACTERISTICS:,,
*24X,4HTYPE,22X,I4,2HMM,1X,2A2,/,
*24X,22HFLASH (VISIBLE) RADIUS,4X,F6.3,1X,6HMETERS,/,
*24X,24HFLASH (VISIBLE) DURATION,2X,F6.3,1X,7HSECONDS)
109 FORMAT(24X,27H*****PROGRAM FLASH END*****,/,21X,40(2H--),1H1)
IWRIT=1
IFLAG=0
C*****READ IN DATA
1 WRITE(IOOUT,100)
CALL DATRD(IWRIT,IFLAG)
IF(ICK.EQ.-1)WAVE=WAVE1
WAVE1=WAVE
IF(IFLAG.EQ.4)GO TO 9999
JSRC=ISRC
IF(ISRC.LT.1.OR.ISRC.GT.2)JSRC=3
FLASHR=FLASH1(JSRC)
FLASHT=FLASH2(JSRC)
WRITE(IOOUT,101)
WRITE(IOOUT,102)TIME
WRITE(IOOUT,103)
WRITE(IOOUT,104)(ROBS(I),RTAR(I),RSRC(I),I=1,3)
FLASH010
FLASH020
*FLASH03
*FLASH040
*FLASH050
FLASH060
FLASH070
FLASH080
FLASH090
FLASH100
FLASH110
FLASH120
FLASH130
FLASH140
FLASH150
FLASH160
FLASH170
FLASH180
FLASH190
FLASH200
FLASH210
FLASH220
FLASH230
FLASH240
FLASH250
FLASH260
FLASH270
FLASH280
FLASH290
FLASH300
FLASH310
FLASH320
FLASH330
FLASH340
FLASH350
FLASH360
FLASH370
FLASH380
FLASH390
FLASH400
FLASH410
FLASH420
FLASH430
FLASH440
FLASH450
FLASH460
FLASH470
FLASH480
FLASH490
FLASH500
FLASH510
FLASH520
FLASH530
FLASH540
FLASH550
FLASH560
FLASH570
FLASH580
FLASH590
FLASH600
FLASH610
FLASH620
FLASH630
FLASH640
FLASH650
FLASH660
FLASH670
FLASH680
FLASH690
FLASH700

```

```

WRITE(IQOUT,105)TARDEC,TIMGUN
WRITE(IQOUT,106)FOV,WAVE,RADSEE,RADLOC,RCTSEE,RCTLOC
WRITE(IQOUT,107)TARLEN,TARWID,TARHGT,PCTSEE,PCTLOC
WRITE(IQOUT,108)IGUN(JSRC),IC1(JSRC),IC2(JSRC),FLASHR,FLASHT
CALL GETIM(FLASHR,FLASHT,ISTOP,ISEE,ILOC,TIMLEF,TIMNOL)
WRITE(IQOUT,98)
CALL DATWT(ISTOP,ISEE,ILOC,TIMLEF,TIMNOL)
WRITE(IQOUT,99)
GO TO 1
9999 WRITE(IQOUT,109)
C*****DEFINE EOSAEL OUTPUT
      ICHK=ISEE+1
      STOP
      END

```

```

FLASH710
FLASH720
FLASH730
FLASH740
FLASH750
FLASH760
FLASH770
FLASH780
FLASH790
FLASH800
FLASH810
FLASH820
FLASH830
FLASH840

```

```

SUBROUTINE DATRD(IWRIT,IFLAG)
C/*****
C/***** SUBROUTINE DATRD
C/***** FLASH MODULE
C/***** EOSAEL80
C/***** THIS SUBROUTINE READS INPUT DATA IN EXACTLY THE SAME FORMAT AS
C/***** THE SMOKE(EOSAEL) AND GRNAD(EOSAEL) MODULES
C/*****
C/***** EACH CARD BEGINS WITH A 4 LETTER IDENTIFIER IN COL 1-4,
C/***** FOLLOWED BY AS MANY (REAL) FIELDS AS NEEDED, 10 COL.
C/***** PER FIELD BEGINNING IN COL 11. THE CARDS ARE NOT ORDER
C/***** DEPENDENT.
C/***** NAME IGNORED
C/***** SCEN
C/***** TIME SCENARIO REFERENCE TIME(SEC)
C/***** ISRC SOURCE TYPE CODE (1=105MM 2=100MM)
C/***** OBSC ROBS(3) OBSERVER COORDINATES (X,Y,Z METERS)
C/***** TARC RTAR(3) TARGET COORDINATES (X,Y,Z METERS)
C/***** TARDEG TARGET ORIENTATION COUNTERCLOCKWISE TO
C/***** POSITIVE X AXIS
C/***** SRCC RSRC(3) COORDINATES OF FLASH CENTER (X,Y,Z METERS)
C/***** TIMGUN TIME OF INIATION OF GUNFLASH
C/***** DCHR
C/***** FOV DETECTOR FIELD OF VIEW (DEGREES)
C/***** WAVE DETECTOR WAVELENGTH (MICROMETERS)
C/***** RADSEE ANGULAR RESOLUTION NEEDED TO DETECT
C/***** (MILLIRADIANS)
C/***** RADLOC ANGULAR RESOLUTION NEEDED TO LOCK ON
C/***** (MILLIRADIANS)
C/***** RCTSEE RECOVERY TIME AT 100 METERS FOR
C/***** DETECTION (SECONDS)
C/***** RCTLOC RECOVERY TIME AT 100 METERS FOR
C/***** LOCK ON (SECONDS)
C/***** TCHR
C/***** TARLEN TARGET LENGTH (METERS)
C/***** TARWID TARGET WIDTH (METERS)
C/***** TARHGT TARGET HEIGHT (METERS)
C/***** PCTSEE FRACTION OF EXPOSURE NEEDED FOR DETECTION
C/***** (PERCENT)
C/***** PCTLOC FRACTION OF EXPOSURE NEEDED FOR LOCK ON
C/***** (PERCENT)
C/***** GO SIGNIFIES END OF THIS RUN, BUT NOT END OF INPUT
C/***** DONE END OF JOB.
C/*****
C/***** COMMON/IOUNT/IOIN,IOOUT
C/***** COMMON/IDATA/ROBS(3),RTAR(3),TARDEG,RSRC(3),TIMGUN,FOV,WAVE,
C/***** RADSEE,RADLOC,TARLEN,TARWID,TARHGT,PCTSEE,PCTLOC,
C/***** TIME,ISRC,RCTSEE,RCTLOC
C/***** DIMENSION IR(18),IR1(2),R1(7),INAME(35)
C/***** DATA IR/2HNA,2HME,2HSC,2HEN,2HOB,2HSC,2HTA,2HRC,2HSR,2HCC,
C/***** 2HDC,2HHR,2HTC,2HHR,2HGO,2H ,2HDO,2HNE/
100 FORMAT(21X,20H*****CARD INPUT*****,/,21X,40(2H--))
101 FORMAT(2A2,6X,7F10.3)
102 FORMAT(21X,2A2,6X,7F10.3)
103 FORMAT(2A2,6X,35A2)
104 FORMAT(21X,2A2,6X,35A2)
C/*****
C/***** BEGINNING OF READ LOOP
C/*****
IF(IWRIT.EQ.0)GO TO 6
WRITE(IOOUT,100)
DO 10 I=1,9
IF(I.EQ.9)GO TO 90
IF(IFLAG.GT.0)GO TO 4
IFLAG=1
READ(IOIN,103)IR1(1),IR1(2),(INAME(J),J=1,35)

```

DTRD0010
DTRD0020
DTRD0030
DTRD0040
DTRD0050
DTRD0060
DTRD0070
DTRD0080
DTRD0090
DTRD0100
DTRD0110
DTRD0120
DTRD0130
DTRD0140
DTRD0150
DTRD0160
DTRD0170
DTRD0180
DTRD0190
DTRD0200
DTRD0210
DTRD0220
DTRD0230
DTRD0240
DTRD0250
DTRD0260
DTRD0270
DTRD0280
DTRD0290
DTRD0300
DTRD0310
DTRD0320
DTRD0330
DTRD0340
DTRD0350
DTRD0360
DTRD0370
DTRD0380
DTRD0390
DTRD0400
DTRD0410
DTRD0420
DTRD0430
DTRD0440
DTRD0450
DTRD0460
DTRD0470
DTRD0480
DTRD0490
DTRD0500
DTRD0510
DTRD0520
DTRD0530
DTRD0540
DTRD0550
DTRD0560
DTRD0570
DTRD0580
DTRD0590
DTRD0600
DTRD0610
DTRD0620
DTRD0630
DTRD0640
DTRD0650
DTRD0660
DTRD0670
DTRD0680
DTRD0690
DTRD0700

IF(IWRIT.EQ.0)GO TO 4	DTRD0710
WRITE(IOOUT,104)IR(1),IR(2),<INAME(J),J=1,35>	DTRD0720
4 READ(IOIN,101) IR(1),IR(2),<R(J),J=1,7>	DTRD0730
IF(IWRIT.EQ.0) GO TO 5	DTRD0740
WRITE(IOOUT,102) IR(1),IR(2),<R(J),J=1,7>	DTRD0750
5 IF(IR(1).EQ.IR(17).AND.IR(2).EQ.IR(18)) GO TO 998	DTRD0760
C*****	DTRD0770
C*** RELATING INPUT DATA TO VARIABLE NAMES.	DTRD0780
C*****	DTRD0790
IF(IR(1).EQ.IR(1).AND.IR(2).EQ.IR(2)) GO TO 10	DTRD0800
IF(IR(1).EQ.IR(3).AND.IR(2).EQ.IR(4)) GO TO 20	DTRD0810
IF(IR(1).EQ.IR(5).AND.IR(2).EQ.IR(6)) GO TO 30	DTRD0820
IF(IR(1).EQ.IR(7).AND.IR(2).EQ.IR(8)) GO TO 40	DTRD0830
IF(IR(1).EQ.IR(9).AND.IR(2).EQ.IR(10)) GO TO 50	DTRD0840
IF(IR(1).EQ.IR(11).AND.IR(2).EQ.IR(12)) GO TO 60	DTRD0850
IF(IR(1).EQ.IR(13).AND.IR(2).EQ.IR(14)) GO TO 70	DTRD0860
IF(IR(1).EQ.IR(15).AND.IR(2).EQ.IR(16)) GO TO 9999	DTRD0870
C*****	DTRD0880
C ERROR CAUTION FOR INVALID DATA CARD	DTRD0890
C*****	DTRD0900
IFLAG=2	DTRD0910
WRITE(IOOUT,105)	DTRD0920
105 FORMAT(21X,35H*****CAUTION***** INVALID DATA CARD)	DTRD0930
GO TO 10	DTRD0940
20 TIME=R(1)	DTRD0950
ISRC=IFIX(R(2))	DTRD0960
GO TO 10	DTRD0970
30 ROBS(1)=R(1)	DTRD0980
ROBS(2)=R(2)	DTRD0990
ROBS(3)=R(3)	DTRD1000
GO TO 10	DTRD1010
40 RTAR(1)=R(1)	DTRD1020
RTAR(2)=R(2)	DTRD1030
RTAR(3)=R(3)	DTRD1040
TARDEG=R(4)	DTRD1050
GO TO 10	DTRD1060
50 RSRC(1)=R(1)	DTRD1070
RSRC(2)=R(2)	DTRD1080
RSRC(3)=R(3)	DTRD1090
TIMGUN=R(4)	DTRD1100
GO TO 10	DTRD1110
60 FOV=R(1)	DTRD1120
WAVE=R(2)	DTRD1130
RADSEE=R(3)	DTRD1140
RADLOC=R(4)	DTRD1150
RCTSEE=R(5)	DTRD1160
RCTLOC=R(6)	DTRD1170
GO TO 10	DTRD1180
70 TARLEN=R(1)	DTRD1190
TARWID=R(2)	DTRD1200
TARHGT=R(3)	DTRD1210
PCTSEE=R(4)	DTRD1220
PCTLOC=R(5)	DTRD1230
10 CONTINUE	DTRD1240
GO TO 9999	DTRD1250
C*****	DTRD1260
C*****CAUTION FOR TOO MANY CARDS	DTRD1270
C*****	DTRD1280
90 WRITE(IOOUT,106)	DTRD1290
IFLAG=3	DTRD1300
106 FORMAT(21X,17H*****CAUTION***** ,/	DTRD1310
*21X,56HMORE THAN 10 DATA CARDS ENTERED---REMAINING CARDS IGNORED)	DTRD1320
GO TO 9999	DTRD1330
998 IFLAG=4	DTRD1340
9999 RETURN	DTRD1350
END	DTRD1360


```

SUBROUTINE DATWT(ISTOP,ISEE,ILOC,TIMLEF,TIMNOL)
C*****
C* SUBROUTINE DATWT
C* FLASH MODULE
C* EOSAEL80
C*****
COMMON/IOUNT/IOIN,IOOUT
COMMON/IDATA/ROBS(3),RTAR(3),TARDEG,RSRC(3),TIMGUN,FOV,WAVE,
* RADSEE,RADLOC,TARLEN,TARWID,TARHGT,PCTSEE,PCTLOC,
* TIME,ISRC,RCTSEE,RCTLOC
DIMENSION I1(3),I2(3)
DATA I1/2HNO,2HYE,2HPA/
DATA I2/2H ,2HS ,2HRT/
IANS1=ISEE+1
IF(ISEE.EQ.3)IANS1=3
IF(ILOC.EQ.0)IANS2=2
IF(ILOC.EQ.1)IANS2=1
IF(ILOC.EQ.2)IANS2=3
IF(ILOC.EQ.0)IANS2=2
IF(ILOC.EQ.3)IANS2=3
82 FORMAT(21X,36HDETECTION EXPOSURE CRITERIA DEFEATED)
83 FORMAT(21X,38HDETECTION RESOLUTION CRITERIA DEFEATED)
92 FORMAT(21X,34HLOCK ON EXPOSURE CRITERIA DEFEATED)
93 FORMAT(21X,36HLOCK ON RESOLUTION CRITERIA DEFEATED)
97 FORMAT(21X,17HTARGET OBSCURED ?,1X,2A2,/,
* 21X,15HTARGET LOCKED ?,1X,2A2,/)
98 FORMAT(21X,10HTIME LEFT:,4X,F8.3,1X,7HSECONDS,/,
* 21X,13HTIME NO LOCK:,1X,F8.3,1X,7HSECONDS)
99 FORMAT(21X,25HPROGRAM FLASH--STOP CODE:,1X,3I1)
100 FORMAT(21X,11HSOURCE TYPE,1X,I2,1X,12HUNIDENTIFIED)
101 FORMAT(21X,17HINPUT WAVELENGTH:,1X,F6.1,1X,11HMICROMETERS,
* 25H IS OUT OF RANGE OF MODEL)
102 FORMAT(21X,37HGUNFLASH HAS NOT OCCURED YET: TIME = ,1X,F8.3,/,42X,
* 16HTIME OF FLASH = ,1X,F8.3)
103 FORMAT(21X,38HFLASH IS NOT IN DETECTOR FIELD OF VIEW)
104 FORMAT(21X,24HFLASH IS BEHIND TARGET--,
* 15H NO OBSCURATION)
105 FORMAT(21X,24HNORMAL PROGRAM EXECUTION)
WRITE(IOOUT,99)ISTOP,ISEE,ILOC
IF(ISTOP.GT.1)GO TO 1
IF(ISTOP.GT.0)GO TO 10
WRITE(IOOUT,100)ISRC
GO TO 9999
10 WRITE(IOOUT,101)WAVE
GO TO 9999
1 WRITE(IOOUT,97)I1(IANS1),I2(IANS1),I1(IANS2),I2(IANS2)
IF(ISEE.LT.2.AND.ILOC.LT.2)GO TO 11
IF(ISEE.EQ.2)WRITE(IOOUT,82)
IF(ISEE.EQ.3)WRITE(IOOUT,83)
IF(ILOC.EQ.2)WRITE(IOOUT,92)
IF(ILOC.EQ.3)WRITE(IOOUT,93)
GO TO 9998
11 IGO=ISTOP-1
GO TO (2,3,4,5,6,7,8,9),IGO
2 WRITE(IOOUT,102)TIME,TIMGUN
GO TO 9999
3 WRITE(IOOUT,103)
GO TO 9999
4 WRITE(IOOUT,103)
GO TO 9999
5 WRITE(IOOUT,104)
GO TO 9999
6 WRITE(IOOUT,103)
GO TO 9999
7 WRITE(IOOUT,103)
GO TO 9999
8 WRITE(IOOUT,103)
GO TO 9999
9 WRITE(IOOUT,105)
9998 WRITE(IOOUT,98)TIMLEF,TIMNOL

```

```

DAT00010
DAT00020
* DAT00070
* DAT00040
* DAT00050
DAT00060
DAT00070
DAT00080
DAT00090
DAT00100
DAT00110
DAT00120
DAT00130
DAT00140
DAT00150
DAT00160
DAT00170
DAT00180
DAT00190
DAT00200
DAT00210
DAT00220
DAT00230
DAT00240
DAT00250
DAT00260
DAT00270
DAT00280
DAT00290
DAT00300
DAT00310
DAT00320
DAT00330
DAT00340
DAT00350
DAT00360
DAT00370
DAT00380
DAT00390
DAT00400
DAT00410
DAT00420
DAT00430
DAT00440
DAT00450
DAT00460
DAT00470
DAT00480
DAT00490
DAT00500
DAT00510
DAT00520
DAT00530
DAT00540
DAT00550
DAT00560
DAT00570
DAT00580
DAT00590
DAT00600
DAT00610
DAT00620
DAT00630
DAT00640
DAT00650
DAT00660
DAT00670
DAT00680
DAT00690
DAT00700

```

9999 RETURN
END

DAT00710
DAT00720

```

SUBROUTINE GETIM(FLASHR,FLASHT,ISTOP,ISEE,ILOC,TIMLEF,TIMNOL)      GTM00010
C*****GTM00020
C*      SUBROUTINE GETIM      *GTM00030
C*      FLASH MODULE      *GTM00040
C*      EOSAEL80      *GTM00050
C*****GTM00060
REAL LOCSTO,LOCFRA      GTM00070
COMMON/IDATA/ROBS(3),RTAR(3),TARDEG,RSRC(3),TIMGUN,FOV,WAVE,      GTM00080
*      RADSEE,RADLOC,TARLEN,TARWID,TARHGT,PCTSEE,PCTLOC,      GTM00090
*      TIME,ISRC,RCTSEE,RCTLOC      GTM00100
COMMON/CONST/PI      GTM00110
C*****ALL DISTANCES IN METERS ANGLES CONVERTED TO RADIANS(CCW A-AXES      GTM00120
DIMENSION DOTVEC(3),RHTVEC(3),RHFVEC(3),RFLASH(3)      GTM00130
C*****SCALAR(A,B)=A(1)*B(1)+A(2)*B(2)+A(3)*B(3)      GTM00140
SCALAR(A1,A2,A3,B1,B2,B3)=A1*B1+A2*B2+A3*B3      GTM00150
C*****DEFAULT AND CONVERT INPUT      GTM00160
ANGGUN=0.0      GTM00170
ELEGUN=0.0      GTM00180
GUNHT=0.0      GTM00190
GUNLEN=0.0      GTM00200
TARHHT=TARHGT/2.0      GTM00210
ANGTAR=TARDEG*(PI/180.0)      GTM00220
HAFFOV=(FOV/2.0)*(PI/180.0)      GTM00230
SEESTO=RADSEE/1000.0      GTM00240
LOCSTO=RADLOC/1000.0      GTM00250
SEEFRA=PCTSEE/100.0      GTM00260
LOCFRA=PCTLOC/100.0      GTM00270
RADFL=FLASHR      GTM00280
DURTIM=FLASHT      GTM00290
DURSEE=10.0*RCTSEE      GTM00300
DURLOC=10.0*RCTLOC      GTM00310
REACFA=SEEFRA-LOCFRA      GTM00320
C*****INITIALIZE FLAGS *****      GTM00330
AGINSE=0.0      GTM00340
AGINLO=0.0      GTM00350
TIMLEF=0.0      GTM00360
TIMNOL=0.0      GTM00370
TIMGUN=TIME-TIMGUN      GTM00380
ISEE=0      GTM00390
ILOC=0      GTM00400
ISTOP=0      GTM00410
IF(ISRC.LT.1.OR.ISRC.GT.2)GO TO 9999      GTM00420
ISTOP=1      GTM00430
IWAVE=0      GTM00440
IF(WAVE.GE.0.40.AND.WAVE.LE.0.70)IWAVE=1      GTM00450
IF(WAVE.GE.8.00.AND.WAVE.LE.12.0)IWAVE=2      GTM00460
IF(IWAVE.EQ.0)GO TO 9999      GTM00470
ISTOP=2      GTM00480
IF(TIMGUN.LT.0.)GO TO 9999      GTM00490
TIMLEF=DURTIM-TIMGUN      GTM00500
C*****TIMLEF IS DURATION LEFT OF FLASH      GTM00510
TIMNOL =TIMLEF+REACFA*DURTIM      GTM00520
C*****TIMNOL IS DURATION OF LOCK      GTM00530
FLSHIF=RADFL      GTM00540
C*****CALCULATE FLASH COORDINATES FROM GIVEN COORDINATES      GTM00550
RFLASH(1)=RSRC(1)+FLSHIF*COS(ANGGUN)*COS(ELEGUN)      GTM00560
RFLASH(2)=RSRC(2)+FLSHIF*SIN(ANGGUN)*COS(ELEGUN)      GTM00570
RFLASH(3)=RSRC(3)+FLSHIF*SIN(ELEGUN)*GUNHT      GTM00580
C*****DEFINE DIRECTION OF TARGET UNIT VECTOR      GTM00590
DOTVEC(1)=COS(ANGTAR)      GTM00600
DOTVEC(2)=SIN(ANGTAR)      GTM00610
DOTVEC(3)=0      GTM00620
C*****DEFINE OBSERVER-TARGET,OBSERVER-FLASH VECTORS      GTM00630
DO 10 I=1,3      GTM00640
RHTVEC(I)=RTAR(I)-ROBS(I)      GTM00650
RHFVEC(I)=RFLASH(I)-ROBS(I)      GTM00660
10 CONTINUE      GTM00670
C*****ADD HALF TARGET HEIGHT      GTM00680
RHTVEC(3)=RHTVEC(3)+TARHHT      GTM00690
C*****FIND LENGTHS      GTM00700

```

```

A1=RHTVEC(1)
A2=RHTVEC(2)
A3=RHTVEC(3)
B1=RHFVEC(1)
B2=RHFVEC(2)
B3=RHFVEC(3)
RHT=SQRT(SCALAR(A1,A2,A3,A1,A2,A3))
RHF=SQRT(SCALAR(B1,B2,B3,B1,B2,B3))
C*****FIND COSINES BETWEEN VECTORS
C1=DOTVEC(1)
C2=DOTVEC(2)
C3=DOTVEC(3)
GAMMAD= SCALAR(C1,C2,C3,B1,B2,B3)/RHF
GAMMAT= SCALAR(A1,A2,A3,B1,B2,B3)/(RHT*RHF)
W= SCALAR(C1,C2,C3,A1,A2,A3)/RHT
C*****FIND TRIG FCNS OF THETA=HALF-ANGLE OF FLASH CONE
SINTH=RADFL/RHF
COSTH=SQRT(1.0-SINTH*SINTH)
COSSQ=COSTH*COSTH
COSINE=ABS(W)
SINE=SQRT(1.0-COSINE**2)
C*****CALCULATE CYLINDRICAL TARGET LONGEST DIMENSION AS SEEN IN
C*****PLANE PERPENDICULAR TO RHFVEC
TARDIM=TARLEN*SINE+TARWID*COSINE
HAFDIM=TARDIM/2.
THETA=ATAN2(SQRT(1.0-COSTH**2),COSTH)
THETAT=ATAN2(SQRT(1.0-GAMMAT**2),GAMMAT)
DELTH=THETAT-THETA
ISTOP=3
IF(DELTH.GT.HAFFOV)GO TO 9999
C*****IF OUT OF FIELD OF VIEW,RETURN.
DISPLC=RHT*SIN(DELTH)
PROJRH=RHT*GAMMAT
ISTOP=4
IF(DISPLC.GT.HAFDIM)GO TO 99
ISTOP=5
IF(PROJRH.LT.RHF.AND.IWAVE.LT.2)GO TO 99
C*****NO OBSCURATION IF TARGET IN FRONT OF FLASH SO RETURN
C*****FROM HERE ON, GAMMAT NECESSARILY POSITIVE
C*****FIND INTERSECTIONS OF FLASH CONE WITH DOTVEC EXTENDED FROM TARGET
C*****GET COEFFICIENT OF QUADRATIC EQUATION FOR DISTANCE ALONG DOTVEC
C*****FROM TARGET TO FLASH CONE/RHT
A=COSSQ-GAMMAD**2
BD2=COSSQ*W-GAMMAD*GAMMAT
C=COSSQ-GAMMAT**2
B=BD2*2.
C*****IF DISCRIMINANT NEGATIVE, NO INTERSECTION
DISCRM=BD2*BD2-A*C
ISTOP=6
IF(DISCRLT.0.)GO TO 99
C*****IF A=0,QUADRATIC FORMULA BLOWS UP.ACTUALLY HAVE LINEAR EQN.
IF(ABS(A).GT.1.E-30)GOTO20
IF(ABS(B).LT.1.E-30)B=SIGN(1.E-30,B)
SPL=-C/B*RHT
ZPL=SPL*GAMMAD+PROJRH
ISTOP=7
IF(ZPL.LE.0.)GO TO 99
C*****REJECT IF SOLE INTERSEC IS W NEG CONE SHEET.MEANINGLESS
DPL=ABS(SPL)*SINE
IF(GAMMAT-COSTH)14,14,12
C*****IF TAR OUTSIDE FLASH CONE GO TO 14,OTHERWISE 12
12 OBSCUR=HAFDIM+AMIN1(DPL,HAFDIM)
GO TO 30
14 OBSCUR=AMAX1(0.,HAFDIM-DPL)
GO TO 30
20 ROOT=SQRT(DISCRL)
ROOT=SIGN(ROOT,A)
C*****ABOVE NOT REALLY NEEDED BUT NICER TO HAVE SPL.GT.SMI-SEE BELOW
SPL=(-BD2+ROOT)/A *RHT
SMI=(-BD2-ROOT)/A *RHT

```

```

GTM00710
GTM00720
GTM00730
GTM00740
GTM00750
GTM00760
GTM00770
GTM00780
GTM00790
GTM00800
GTM00810
GTM00820
GTM00830
GTM00840
GTM00850
GTM00860
GTM00870
GTM00880
GTM00890
GTM00900
GTM00910
GTM00920
GTM00930
GTM00940
GTM00950
GTM00960
GTM00970
GTM00980
GTM00990
GTM01000
GTM01010
GTM01020
GTM01030
GTM01040
GTM01050
GTM01060
GTM01070
GTM01080
GTM01090
GTM01100
GTM01110
GTM01120
GTM01130
GTM01140
GTM01150
GTM01160
GTM01170
GTM01180
GTM01190
GTM01200
GTM01210
GTM01220
GTM01230
GTM01240
GTM01250
GTM01260
GTM01270
GTM01280
GTM01290
GTM01300
GTM01310
GTM01320
GTM01330
GTM01340
GTM01350
GTM01360
GTM01370
GTM01380
GTM01390
GTM01400

```

C*****SPL,SMI ARE 2 LENGTHS ALONG DOTVEC OF INTERSEC PTS W FLASH CONE	GTM01410
C*****ZPL,ZMI ARE 2 COORDINATES OF INTERSECTIONS<ON FLASH CONE AXIS>	GTM01420
ZPL=SPL*GAMMAD+PROJRH	GTM01430
ZMI=SMI*GAMMAD+PROJRH	GTM01440
ISTOP=8	GTM01450
IF<ZPL.LE.0..AND.ZMI.LE.0.>GO TO 99	GTM01460
C*****REJECT IF BOTH INTERCEPTS IN NEGATIVE CONE	GTM01470
C*****DPL,DMI ARE PROJECTIONS OF SPL,SMI PERPEN TO RHTVEC	GTM01480
C*****THEIR MAGNITUDES ARE LIMITED TO HAFDIM	GTM01490
DPL=SPL*SINE	GTM01500
DPL=AMIN1<DPL,HAFDIM>	GTM01510
DPL=AMAX1<DPL,-HAFDIM>	GTM01520
DMI=SMI*SINE	GTM01530
DMI=AMIN1<DMI,HAFDIM>	GTM01540
DMI=AMAX1<DMI,-HAFDIM>	GTM01550
OBSCUR=ABS<DPL-DMI>	GTM01560
IF<ZPL*ZMI.GT.0.>GOTO30	GTM01570
C*****SKIP AROUND UNLESS BOTH SHEETS OF CONE INVOLVED	GTM01580
IF<ZPL.LE.0.>DPL=SIGN<HAFDIM,DPL>	GTM01590
IF<ZMI.LE.0.>DMI=SIGN<HAFDIM,DMI>	GTM01600
OBSCUR=TARDIM-ABS<DPL-DMI>	GTM01610
C*****ABOVE BRANCH RARE INTERSEC W BOTH CONE SHEETS.NEG SHEET IGNORE.	GTM01620
30 CONTINUE	GTM01630
SEEN=TARDIM-OBSCUR	GTM01640
ISTOP=9	GTM01650
98 IF<IWAVE.EQ.2>GO TO 1	GTM01660
CALL VSBLC<ISEE,ILOC,SEEN,TARDIM,RHT,TIMLEF,SEESTO,LOCSTO,	GTM01670
* SEEFRA,LOCFRA>	GTM01680
GO TO 99	GTM01690
1 CALL IRBLC<RHF,ISEE,ILOC,AGINSE,AGINLO,DURSEE,DURLOC,TIMGON>	GTM01700
99 CONTINUE	GTM01710
TIMLEF=AMAX1<TIMLEF,AGINSE>	GTM01720
TIMNOL=AMAX1<TIMNOL,AGINLO>	GTM01730
9999 RETURN	GTM01740
END	GTM01750

```

SUBROUTINE IRBLC(RHF, ISEE, ILOC, AGINSE, AGINLO, DURSEE, DURLOC, TIMGON) IRBLC010
C***** IRBLC020
C* SUBROUTINE IRBLC * IRBLC030
C* FLASH MODULE * IRBLC040
C* EDSAE180 * IRBLC050
C***** IRBLC060
INTEGER LOCIRM IRBLC070
SEEIRM=4.0 IRBLC080
LOCIRM=5.0 IRBLC090
SQRTRH=SQRT(RHF) IRBLC100
AGINSE=DURSEE/SQRTRH IRBLC110
AGINLO=DURLOC/SQRTRH IRBLC120
C***** ABOVE WHOLLY EMPIRICAL FROM FIT TO TV TAPES AT 2 RANGES IRBLC130
IF(AGINSE.GT.SEEIRM) AGINSE=SEEIRM IRBLC140
IF(AGINLO.GT.LOCIRM) AGINLO=LOCIRM IRBLC150
IF(AGINSE.GT.TIMGON) ISEE=1 IRBLC160
IF(AGINLO.GT.TIMGON) ILOC=1 IRBLC170
RETURN IRBLC180
END IRBLC190

```

SUBROUTINE VSBLC(ISEE, ILOC, SEEN, TARDIM, RHT, TIMLEF, SEESTO, LOCSTO,	VSBLC010
* SEEFRA, LOCFRA)	VSBLC020
C*****	VSBLC030
C* SUBROUTINE VSBLC	* VSBLC040
C* FLASH MODULE	* VSBLC050
C* EOSAEL80	* VSBLC060
C*****	VSBLC070
REAL LOCCRI, LOCFRA, LOCSTO	VSBLC080
C*****SEEFRA IS FRACTION THAT MUST BE SEEN, LIKEWISE FOR LOCK	VSBLC090
C*****SEESTO IS MIN ANGLE THAT MUST BE SEEN, LIKEWISE FOR LOCK	VSBLC100
SEEANG=SEEN/RHT	VSBLC110
TARANG=TARDIM/RHT	VSBLC120
C*****DEFINE SEEN ANGLE AND TARGET ANGLE	VSBLC130
SEECRI=AMIN1(TARANG, SEESTO)	VSBLC140
LOCCRI=AMIN1(TARANG, LOCSTO)	VSBLC150
IF(TIMLEF.LE.0.)GO TO 9999	VSBLC160
IF(SEEANG.LT.SEEFRA*TARANG)GO TO 48	VSBLC170
ISEE=2	VSBLC180
GO TO 50	VSBLC190
48 IF(SEEANG.LT.SEECRI)GO TO 49	VSBLC200
ISEE=1	VSBLC210
GO TO 50	VSBLC220
49 CONTINUE	VSBLC230
ISEE=3	VSBLC240
50 CONTINUE	VSBLC250
IF(SEEANG.LT.LOCFRA*TARANG)GO TO 98	VSBLC260
ILOC=2	VSBLC270
GO TO 9999	VSBLC280
98 IF(SEEANG.LT.LOCCRI)GO TO 99	VSBLC290
ILOC=1	VSBLC300
GO TO 9999	VSBLC310
99 CONTINUE	VSBLC320
ILOC=3	VSBLC330
9999 RETURN	VSBLC340
END	VSBLC350

EORUN	, 4.0								
WAVL	, 10.6	10.6							
VIS	, 7.0								
TURB									
XSCALE	, 3.0								
SNOKE									
DRTRAN									
LZTRAN									
CLTRAN	, 3.0								
SCREEN									
UVRCST	, 2.0								
CLIMAT	, 1.0	1.0	5.0	6.0	-1.0	1.0			
GRNADE									
GO									
PARM	0.1016	1.3E-04	0.0	400.0	5.0	512.0			
CN1	1.0	6.0E-13	1.0E-14	1.3E-14	2.7E-14	5.0E-14			
V1	1.0	0.93	0.93	0.93	0.93	0.93			
DVRV	0.400	500.0							
CN2	1.0	6.0E-13	1.0E-14	1.3E-14	2.7E-14	5.0E-14			
V2	1.0	0.93	0.93	0.93	0.93	0.93			
GO									
END									
FOG	1.								
HORZ	0.4								
GO									
FOG	2.								
SLNH	0.133,56.3								
GO									
FOG	3.								
HORZ	0.4								
GO									
END									
MUNC	0.0	-50.0	0.						
OBSC	200.	0.	2.0						
TARC	-200.	0.	50.0						
BART	5.	180.	5.	90.					
OUTP	0.	0.	0.	0.					
BURN	10.	0.	0.	0.	0.				
GO									
BURN	4.	0.	0.	0.	0.				
BART	5.	250.	5.	90.					
GO									
DONE									
END									
MET1	4.	2.	286.0	2.	2.0	0.0			
MET2	1.	53.							
SOIL	2.	0.0	0.15						
CHAR	3.	6.8	0.0						
EXPL	1.	1.	1.	0.0	0.0	0.0	0.0	0.0	0.0
OBSC	200.0	0.0	2.0						
TRNC	-200.0	0.0	2.0						
RECC	200.0	0.0	2.0						
TIMS	1.0	71.0	2.0						
GO	1.								
DONE									
END									
END	8.55	15.	0.4						
SEEK,	0.2,	0.0,	0.6						
TARG,	-0.2,	0.0,	0.002						
CLST,	0.20,	0.40							
GO									
SEEK,	0.133,	0.0,	0.5						
GO									
SEEK,	0.0,	0.0,	0.3						
GO									
END									
1 1 1									
TARV	1.0	2.0	0.24	2.3	81.	2.	.000		
SENS	.99	8.	1.	0.	0.	1.			

GO							
SENS	.90	8.	1.	0.	0.	1.	
GO							
SENS	.75	8.	1.	0.	0.	1.	
GO							
SENS	.50	8.	1.	0.	0.	1.	
GO							
SENS	.10	8.	1.	0.	0.	1.	
GO							
DONE							
SCRN	15.	400.	0.4	0.	90.	74.	2.
DONE							
END							
OPOS,	-0.0667,	0.0,	0.2				
CLDS,	0.2,	200.0,	40.0,	0.7,	1.0		
SPOS,	-0.2,	0.0,	0.002				
BKGR,	50.0						
GRND,	50.0						
TEMP,	9.8						
GO							
UPOS,	-0.1333,	0.0,	0.1				
GO							
END							
NAME							
OUTP							
OBSC	-200.0	0.0	2.0				
MUNC	-200.0	0.0	2.0	95.0	100.0	10.0	
TARC	-200.0	+40.0	50.0				
BART	5.0	400.0	50.0	90.0			
MUNT	1.0	0.793	14.3	1.0	0.0	4.7	0.07
METR	50.0	2.0	220.0	4.0	20.0	0.0	0.0
EXTC							
BURN							
MISC							
GO							
DONE							
END							
WAVL	1.06	1.06					
VIS	5.0	5.0	4.0				
BASCAT							
FCLLOUD							
GO							
PART,	1.	5000.	1.				
SORC,	-0.2,	0.	-0.098,	90.,	0.,	50.,	
DETR,	0.2,	0.	-0.098,	90.,	180.,	1.,	1.
CLDS,	.1,	.2,	.1,	0.,	0.,	0.,	5.
GRND,	-0.1,	0.5					
PULS,	.33,	0.					
GO							
END							
CPUS,	0.0,	0.0,	0.1				
RFOS,	0.2,	0.0,	0.002				
SPOS,	-0.2,	0.0,	0.002				
AXES,	0.1,	0.2,	0.1				
CLDS,	5.0,	0.95,	4.0,	1.0,	9.8		
ATMO,	2.0,	9.8					
BKGR,	0.5,	50.0					
SANG,	80.0,	0.0					
LUND,	0.0						
GO							
END							
FREQ	35.0	35.0					
NMMW	2.0						
GO							
PATH,	0.4						
ATMO,	15.0	1013.25	6.44				
FOGD,	0.5						
RAIN,	5.0						
GO							
ATMO,	-1.0	1015.2	7.7				

RAIN,	0.0								
SNOW,	5.0								
GO									
END									
WVNUM	2010.	2710.	2.						
LT4M									
RESF									
SPOT									
GO									
ENVR	3.	2.	2.	4.	1.	1.			
EMIS	.100+01	.283+03	.950+00	.295+03					
ATM	.650+02	0.0	0						
TARG	.400+00	.450+02	.900+02	.450+02					
REFL	0.5	0.5	0	.500-01	.0	0.			
SENS	.200-02	.900+02	.270+03	.100+01					
GO									
12									
	3.5	0.78							
	3.6	0.83							
	3.7	0.87							
	3.8	0.92							
	3.9	0.96							
	4.0	0.98							
	4.1	0.97							
	4.2	0.96							
	4.3	0.95							
	4.4	0.94							
	4.5	0.93							
	4.6	0.93							
END									
4	2	1	0	1	0	0	0	0.000	0.000
	0.002			0.002		0.000		0.400	0.000
4									1.000
4	2	1	0	1	0	0	1	0.000	0.000
0									
END									

INDIVIDUAL MODULES SELECTED

MODULES SELECTED

ENDING
943.396
10.600
8301.886

BEGINNING
943.396
10.600
28301.886

```
WAVENUMBER(CH** -1)
WAVELENGTH(MICRONS)
FREQUENCY(GHZ)
```

CLIMATOLOGY MODEL

DEFINITIONS OF METEOROLOGICAL CLASSES

CLASS 1	= FOG, HAZE AND MIST WITH VIS LT 1 KM.
CLASS 2	= FOG, HAZE AND MIST WITH 1 LE VIS LT 3 KM.
CLASS 3	= FOG, HAZE AND MIST WITH 3 LE VIS LT 7 KM.
CLASS 4	= FOG, HAZE AND MIST WITH VIS GE 7 KM.
CLASS 5	= DUST WITH VIS LT 1 KM.
CLASS 6	= DUST WITH 1 LE VIS LT 3 KM.
CLASS 7	= DUST WITH 3 LE VIS LT 7 KM.
CLASS 8	= DUST WITH VIS GE 7 KM.
CLASS 9	= DRIZZLE, RAIN AND TSTMS WITH VIS LT 1 KM.
CLASS 10	= DRIZZLE, RAIN AND TSTMS WITH 1 LE VIS LT 3 KM.
CLASS 11	= DRIZZLE, RAIN AND TSTMS WITH 3 LE VIS LT 7 KM.
CLASS 12	= DRIZZLE, RAIN AND TSTMS WITH VIS GE 7 KM.
CLASS 13	= SNOW WITH VIS LT 1 KM.
CLASS 14	= SNOW WITH 1 LE VIS LT 3 KM.
CLASS 15	= SNOW WITH 3 LE VIS LT 7 KM.
CLASS 16	= SNOW WITH VIS GE 7 KM.
CLASS 17	= NO WEATHER AND ABSOLUTE HUMIDITY LT 10 GM/CCU M.
CLASS 18	= NO WEATHER AND ABSOLUTE HUMIDITY GE 10 GM/CCU M.
CLASS 19	= VIS LT 1 KM AND CEILING HEIGHT LT 300 M.
CLASS 20	= VIS LT 3 KM AND CEILING HEIGHT LT 1000 M.
CLASS 21	= NO CEILING.
CLASS 22	= ALL CONDITIONS COMBINED.

EDSHEL CLIMATOLOGY FOR EUROPEAN LOWLANDS									
CLASS NO.	FREQUENCY CLASS (%)	MEAN TEMP (°C)	MEAN DEW (°C)	MEAN RH (CM/CM) (H)	MEAN PRESS (MB)	MEAN WINDVEL (MPS)	MEAN CLOUDT (%)	MEAN CLOUDVR (%)	CLASS NO.
1	10	10	10	10	10	10	10	10	1
2	20	20	20	20	20	20	20	20	2
3	30	30	30	30	30	30	30	30	3
4	40	40	40	40	40	40	40	40	4
5	50	50	50	50	50	50	50	50	5
6	60	60	60	60	60	60	60	60	6
7	70	70	70	70	70	70	70	70	7
8	80	80	80	80	80	80	80	80	8
9	90	90	90	90	90	90	90	90	9
10	100	100	100	100	100	100	100	100	10

TUPB LASER MODULE

CALCULATION OF POWER SPECTRUM AND TURBULENCE INDUCED POINTING JITTER OF A LASER TARGET DESIGNATOR AND SEEKER

LASER WAVELENGTH (MICROMETERS)	10.6000
DESIG. APERTURE DIAMETER (METERS)	.101600
BEAMSPREAD ANGLE (RADIAN)	.000130
SEEKER APERTURE DIAMETER (METERS)	.400000
RANGE FROM TARGET TO SEEKER (METERS)	500.00
BEAM SLUE RATE (RAD/SEC)	.000000
DESIGNATION RANGE (METERS)	400.00
DURATION OF TEST (SECONDS)	5.0000
TOTAL DESIGNATOR PATH SEGMENTS	5
TOTAL SEEKER PATH SEGMENTS	5
TOTAL FREQUENCIES FOR WHICH POWER SPECTRUM IS TO BE CALCULATED	512

VALUES OF REFRACTIVE INDEX STRUCTURE CONSTANT AND WIND SPEED IN DESIGNATOR PATH

SEGMENT NO.	CN**2 (METER**(-2/3))	WINDSPEED (METER/SEC)
1	.600000-012	.93
2	.100000-013	.93
3	.130000-013	.93
4	.270000-013	.93
5	.500000-013	.93

VALUES OF REFRACTIVE INDEX STRUCTURE CONSTANT AND WIND SPEED IN SEEKER PATH

SEGMENT NO.	CN**2 (METER**(-2/3))	WINDSPEED (METER/SEC)
1	.600000-012	.93
2	.100000-013	.93
3	.130000-013	.93
4	.270000-013	.93
5	.500000-013	.93

DESIGNATOR TO TARGET	
VIRTUAL POINT SOURCE TO APERTURE DISTANCE	781.53845 (METERS)
DISTANCE FROM VIRTUAL POINT SOURCE TO TARGET	1181.53845 (METERS)
INTEGRATED COHERENCE LENGTH	.335957 (METERS)
DIAMETER/INTEGRATED COHERENCE LENGTH	.302419
TRANSMITTER-INDUCED BEAM SPREAD	.13000-003 (RADIAN)
DIFFRACTION-LIMITED BEAM SPREAD	.11769-003 (RADIAN)
DIFFRACTION AND TURBULENCE BEAM SPREAD	.11907-003 (RADIAN)
TOTAL EFFECTIVE BEAM SIZE	.17629-003 (RADIAN)
	.172116 (METERS)

SEGMENT NO.	COHERENCE LENGTH	REFERENCE FREQUENCY (HERTZ)
1	341660	56.285500
2	5115430	10.267600
3	6158230	11.267400
4	13.656743	8.125533
5		6.725533

TARGET TO SEEKER

INTEGRATED COHERENCE LENGTH .934983 (METERS)
 DIAMETER/INTEGRATED COHERENCE LENGTH .184085
 TRANSMITTER-INDUCED BEAM SPREAD .13000-003 (RADIAN) .065000 (METERS)
 DIFFRACTION-LIMITED BEAM SPREAD .69469-004 (RADIAN) .034735 (METERS)
 DIFFRACTION AND TURBULENCE BEAM SPREAD .69828-004 (RADIAN) .034914 (METERS)
 TOTAL EFFECTIVE BEAM SIZE .14757-003 (RADIAN) .245906 (METERS)

SEGMENT NO. COHERENCE LENGTH REFERENCE FREQUENCY (HERTZ)

1 2.689629 7.400705
 2 10.458227 2.466902
 3 5.360999 1.489141
 4 2.469819 1.052244
 5 1.327262 1.822301

THE VARIANCE OF THE POWER SPECTRUM IS .8090-011
 OUTPUT FOR DESIGNATOR TO TARGET PATH

MEAN AND VARIANCE OF RANDOM ARRAY

MEAN OF REAL PART = .92021-007, MEAN OF IMAG PART = .00000
 VAR. OF REAL PART = .83284-011, VAR. OF IMAG PART = .00000

MEAN AND VARIANCE OF TIME SEQUENCE

MEAN OF REAL PART = -.52602-011, MEAN OF IMAG PART = .19830-016
 VAR. OF REAL PART = .79425-017, VAR. OF IMAG PART = .13359-031

OUTPUT FOR TARGET TO SEEKER PATH

MEAN AND VARIANCE OF RANDOM ARRAY

MEAN OF REAL PART = .17265-006, MEAN OF IMAG PART = .00000
 VAR. OF REAL PART = .91505-011, VAR. OF IMAG PART = .00000

MEAN AND VARIANCE OF TIME SEQUENCE

MEAN OF REAL PART = .24531-011, MEAN OF IMAG PART = .26733-016
 VAR. OF REAL PART = .87592-017, VAR. OF IMAG PART = .12492-031

XSCALE HORIZONTAL-SLANT PATH EXTINCTION MODULE

OPTIONS CHOSEN
MARITIME ARTIC
HORIZONTAL PATH

EXTINCTION FROM 8.0 TO 12.0 MICRONS	DISTANCE	TRANSMISSION
KM**--1 .178	KM .400	.931+000

OPTIONS CHOSEN
MARITIME POLAR
SLANT PATH FOR 10.600 MICRONS

WARNING FROM SLANT
THE VERTICAL DISTANCE EXCEEDS THE 500 METER UPPER LIMIT, OR
IS NOT AN INTEGER MULTIPLE OF 20 METERS
SLANT DISTANCE CHANGED FROM .2397 TO .2238 KM

SLANT EXTINCTION AT 10.60 MICRONS	DISTANCE	TRANSMISSION	ANGLE
KM**--1 1.295	KM .224	.748+000	56.30

OPTIONS CHOSEN
CONTINENTAL POLAR
HORIZONTAL PATH

EXTINCTION FROM 8.0 TO 12.0 MICRONS	DISTANCE	TRANSMISSION
KM**--1 .008	KM .400	.997+000

SMOKE MODEL MODULE

 * SMOKE *

EXECUTION 1

SMOKE MUNITIONS
 WHITE PHOSPHORUS (WP)
 NO. ROUNDS 1
 FILL WEIGHT 15.600 LB
 BURN TIME 1.0 SEC
 EFFICIENCY 100.0 PERCENT
 YIELD FACTOR 5.84

METEOROLOGICAL CONDITIONS
 WINDSPEED 3.6 M/S
 WIND DIRECTION (USUAL)
 MET CONVECTION AZIMUTH 225.0 DEGREES
 RELATIVE HUMIDITY 87.1 PERCENT
 PASQUILL CATEGORY 0
 AIR TEMPERATURE 5.3 DEGREE C
 TEMP. GRADIENT .00 C DEG./M

EXTINCTION COEFFICIENTS
 0.4-0.7 MICROMETERS 4.304 M**2/GM
 0.7-1.2 MICROMETERS 2.166 M**2/GM
 1.06 MICROMETERS 1.541 M**2/GM
 3.0-5.0 MICROMETERS 1.350 M**2/GM
 8.0-12. MICROMETERS 1.368 M**2/GM
 10.6 MICROMETERS 1.364 M**2/GM
 94.0 GHZ .001 M**2/GM

BURN RATE PROFILE = 1.0000

FIELD COORDINATES
 MUNITION COORDINATES= (X) .00 (Y) -50.00 (Z) .00 METERS
 OBSERVER COORDINATES= 200.00 .00 2.00 METERS
 TARGET COORDINATES= -200.00 .00 2.00 METERS
 ANGLE OF ORIGINAL X-AXIS, CLOCKWISE WRT NORTH = 90.00 DEG.
 EVENT TIME = .0 SEC

ROTATED COORD. (WIND X-AXIS, MUNITION ORIGIN)
 (XW) (YW) (ZW)
 .00 .00 .00 METERS
 176.78 -106.07 2.00 METERS
 -106.07 176.78 2.00 METERS

TIME (SEC)	LENGTH (METERS)	WIDTH (METERS)	HEIGHT (METERS)	PATHLENGTH (METERS)	CL (GM/M**2)	TRANSMISSION SPECTRAL BANDS (MICROMETERS)						
						0.4-0.7	0.7-1.2	1.06	3.0-5.0	8.0-12.	10.6	94.0GHZ
5.	19.	13.	22.	.00	.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000
10.	36.	16.	34.	.00	.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000
15.	54.	20.	44.	.00	.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000
20.	72.	24.	53.	.00	.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000
25.	90.	28.	61.	15.16	4.01	.000	.000	.002	.245	.258	.232	.996
30.	108.	31.	69.	40.05	4.89	.000	.000	.001	.181	.191	.169	.995
35.	126.	34.	76.	45.28	4.48	.000	.001	.005	.296	.309	.282	.997
40.	144.	38.	83.	45.86	1.99	.000	.004	.019	.400	.418	.391	.997
45.	162.	41.	90.	48.25	1.58	.001	.013	.047	.498	.510	.485	.998
50.	180.	43.	95.	50.00	.004	.004	.033	.088	.575	.586	.563	.998
55.	198.	45.	99.	51.65	.007	.004	.062	.138	.638	.648	.626	.999
60.	216.	47.	101.	53.29	.010	.007	.092	.193	.688	.697	.676	.999
65.	234.	49.	111.	54.84	.021	.021	.148	.258	.730	.738	.708	.999
70.	252.	51.	115.	56.30	.036	.036	.236	.358	.764	.771	.739	.999
75.	270.	53.	120.	57.68	.057	.057	.353	.502	.798	.804	.769	.999
80.	288.	55.	124.	59.00	.081	.081	.503	.687	.821	.829	.789	.999
85.	306.	57.	127.	60.27	.109	.109	.683	.902	.840	.846	.806	.999
90.	324.	59.	131.	61.50	.139	.139	.902	1.155	.856	.861	.821	1.000
95.	342.	61.	135.	62.60	.171	.171	1.155	1.448	.870	.874	.834	1.000
100.	360.	63.	138.	63.58	.205	.205	1.448	1.776	.882	.886	.846	1.000
105.	378.	65.	141.	64.45	.241	.241	1.776	2.140	.892	.896	.856	1.000
110.	396.	67.	144.	65.21	.279	.279	2.140	2.540	.902	.906	.866	1.000
115.	414.	69.	147.	65.87	.319	.319	2.540	2.976	.912	.916	.876	1.000
120.	432.	71.	150.	66.44	.361	.361	2.976	3.448	.920	.924	.886	1.000
125.	450.	73.	153.	66.93	.404	.404	3.448	3.956	.928	.932	.896	1.000
130.	468.	75.	155.	67.35	.448	.448	3.956	4.500	.936	.940	.906	1.000
135.	486.	77.	158.	67.71	.493	.493	4.500	5.076	.944	.948	.916	1.000
140.	504.	79.	160.	68.01	.539	.539	5.076	5.692	.952	.956	.926	1.000
145.	522.	81.	163.	68.26	.586	.586	5.692	6.348	.960	.964	.936	1.000
150.	540.	83.	165.	68.46	.634	.634	6.348	7.044	.968	.972	.946	1.000
155.	558.	85.	167.	68.61	.683	.683	7.044	7.780	.976	.980	.956	1.000
160.	576.	87.	170.	68.72	.733	.733	7.780	8.556	.984	.988	.966	1.000
165.	594.	89.	172.	68.79	.783	.783	8.556	9.372	.992	.996	.976	1.000
170.	612.	91.	174.	68.83	.833	.833	9.372	10.228	.999	.999	.984	1.000
175.	630.	93.	176.	68.84	.883	.883	10.228	11.124	.999	.999	.992	1.000
180.	648.	95.	178.	68.83	.933	.933	11.124	12.060	.999	.999	.999	1.000

EXECUTION 2

METEOROLOGICAL CONDITIONS

WINDSPEED	3.6	M/S
WIND DIRECTION (USUAL		
MET CONVENTION AZIMUTH)	225.0	DEGREES
RELATIVE HUMIDITY	87.1	PERCENT
PASQUILL CATEGORY	0	
AIR TEMPERATURE	5.3	DEGREE C
TEMP. GRADIENT	.00	C DEG./M

EXTINCTION COEFFICIENTS			
0.4-0.7	MICROMETERS	4.575	M++GN
0.7-1	MICROMETERS	1.186	M++GN
1.06	MICROMETERS	0.940	M++GN
3.0-5.0	MICROMETERS	1.190	M++GN
8.0-12.	MICROMETERS	.052	M++GN
10.6	MICROMETERS	.051	M++GN
44.0	GHZ	.001	M++GN

BURN RATE PROFILE = .5370+ .4760 (T/TBURN) + 4.7790 (T/TBURN)**2 + -5.4720 (T/TBURN)**3

FIELD COORDINATES			ROTATED COORD. (WIND X-AXIS, MUNITION ORIGIN)			
	(X)	(Y)	(Z)	(XM)	(YM)	(ZM)
MUNITION COORDINATES=	.00	-50.00	.00 METERS	.00	.00	.00 METERS
OBSERVER COORDINATES=	200.00	.00	2.00 METERS	176.78	-106.07	2.00 METERS
TARGET COORDINATES=	-200.00	.00	2.00 METERS	-106.07	176.78	2.00 METERS
ANGLE OF ORIGINAL X-AXIS, CLOCKWISE WRT NORTH =	90.00 DEG.					
EVENT TIME =	.0 SEC					

[illegible]

TIME (SEC)	CL (G/M**2)	COMBINED EFFECT OF 2 EXECUTIONS IN SMOKE:						
		0.4-0.7	0.7-1.2	1.0-1.6	1.6-3.0	3.0-5.0	5.0-10.0	10.0-94.0
0.0	0.0	1.000	1.000	1.000	1.000	1.000	1.000	1.000
0.1	0.0	1.000	1.000	1.000	1.000	1.000	1.000	1.000
0.2	0.0	.949	.975	.977	.998	.999	.999	1.000
0.3	0.0	.000	.000	.000	.374	.764	.768	.995
0.4	0.0	.000	.000	.000	.370	.699	.704	.993
0.5	0.0	.000	.000	.000	.333	.740	.745	.994
0.6	0.0	.000	.000	.000	.271	.763	.767	.995
0.7	0.0	.000	.000	.000	.385	.722	.726	.995
0.8	0.0	.000	.000	.000	.393	.774	.778	.995
0.9	0.0	.000	.000	.000	.385	.770	.774	.995
1.0	0.0	.000	.000	.000	.374	.764	.768	.995
1.1	0.0	.000	.000	.000	.360	.756	.760	.994
1.2	0.0	.000	.000	.000	.347	.748	.752	.994
1.3	0.0	.000	.000	.000	.333	.741	.745	.994
1.4	0.0	.000	.000	.000	.322	.736	.740	.994
1.5	0.0	.000	.000	.000	.311	.734	.738	.994
1.6	0.0	.000	.000	.000	.300	.734	.738	.994
1.7	0.0	.000	.000	.000	.289	.734	.738	.994
1.8	0.0	.000	.000	.000	.278	.734	.738	.994
1.9	0.0	.000	.000	.000	.267	.734	.738	.994
2.0	0.0	.000	.000	.000	.256	.734	.738	.994
2.1	0.0	.000	.000	.000	.245	.734	.738	.994
2.2	0.0	.000	.000	.000	.234	.734	.738	.994
2.3	0.0	.000	.000	.000	.223	.734	.738	.994
2.4	0.0	.000	.000	.000	.212	.734	.738	.994
2.5	0.0	.000	.000	.000	.201	.734	.738	.994
2.6	0.0	.000	.000	.000	.190	.734	.738	.994
2.7	0.0	.000	.000	.000	.179	.734	.738	.994
2.8	0.0	.000	.000	.000	.168	.734	.738	.994
2.9	0.0	.000	.000	.000	.157	.734	.738	.994
3.0	0.0	.000	.000	.000	.146	.734	.738	.994
3.1	0.0	.000	.000	.000	.135	.734	.738	.994
3.2	0.0	.000	.000	.000	.124	.734	.738	.994
3.3	0.0	.000	.000	.000	.113	.734	.738	.994
3.4	0.0	.000	.000	.000	.102	.734	.738	.994
3.5	0.0	.000	.000	.000	.091	.734	.738	.994
3.6	0.0	.000	.000	.000	.080	.734	.738	.994
3.7	0.0	.000	.000	.000	.069	.734	.738	.994
3.8	0.0	.000	.000	.000	.058	.734	.738	.994
3.9	0.0	.000	.000	.000	.047	.734	.738	.994
4.0	0.0	.000	.000	.000	.036	.734	.738	.994
4.1	0.0	.000	.000	.000	.025	.734	.738	.994
4.2	0.0	.000	.000	.000	.014	.734	.738	.994
4.3	0.0	.000	.000	.000	.003	.734	.738	.994
4.4	0.0	.000	.000	.000	.000	.734	.738	.994
4.5	0.0	.000	.000	.000	.000	.734	.738	.994
4.6	0.0	.000	.000	.000	.000	.734	.738	.994
4.7	0.0	.000	.000	.000	.000	.734	.738	.994
4.8	0.0	.000	.000	.000	.000	.734	.738	.994
4.9	0.0	.000	.000	.000	.000	.734	.738	.994
5.0	0.0	.000	.000	.000	.000	.734	.738	.994

***TRANSMISSION RETURNED TO MAIN FOR WAVELENGTH OF 10.600 MICROMETERS IS .999 AT TIME 250.

DIRT TRANSMISSION MODULE

DIRTRAN-2 DUST CLOUD INFRARED TRANSMISSION CALCULATION

*** NOTE -- ALL UNITS ARE MKS UNLESS OTHERWISE SPECIFIED ***

HT 2.00 PASQUILL CATEGORY D
 TEMP 278.36 HT 2.00 WIND 3.60
 WIND DIRECTION 45.00
 LATITUDE 53.00
 THE INVERSION LAYER HEIGHT IS GROWING
 SOIL-2
 SILT CONTENT .15
 DEPTH OF SOIL .00
 30 DEGREE TILTED TIP AT 0.3 METER DEPTH
 WEIGHT OF CHARGE 6.80 KG.
 DETONATION DEPTH .00
 SIMULTANEOUS BURST, UNIFORMLY DISTRIBUTED CHARGES IN A PARALLELOGRAM
 TOTAL NUMBER OF CHARGES IS 1 WITH REFERENCE CHARGE AT (.00, .00) .00
 1 CHARGES WITH DIRECTION AND SPACING GIVEN BY (.00, .00)
 1 CHARGES WITH DIRECTION AND SPACING GIVEN BY (.00, .00)

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 1.00

WAVELENGTH 10.60 MICROMETERS
 TRANSMITTER COORDINATES -200.00 .00 2.00
 RECEIVER COORDINATES 200.00 .00 2.00
 TRANSMITTANCE ALONG THE LINE OF SIGHT .767-002

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00
 THE HEIGHT OF THE CLOUD IS 19.74 METERS
 THE CENTROID COORDINATES ARE 1.34 10.55
 THE WIDTH AT THE CENTROID IS 20.62 METERS
 THE WIDTH AT 2.00 METERS IS 16.56 METERS
 6 CONTOUR POINTS HAVE BEEN DETERMINED
 -5.670 2.000
 -2.973 10.550
 1.337 19.746
 11.646 10.550
 10.693 2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 3.00

WAVELENGTH 10.60 MICROMETERS
 TRANSMITTER COORDINATES -200.00 .00 2.00
 RECEIVER COORDINATES 200.00 .00 2.00
 TRANSMITTANCE ALONG THE LINE OF SIGHT .626+000

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00
 THE HEIGHT OF THE CLOUD IS 21.89 METERS
 THE CENTROID COORDINATES ARE 8.02 10.55
 THE WIDTH AT THE CENTROID IS 31.26 METERS
 THE WIDTH AT 2.00 METERS IS 21.88 METERS
 6 CONTOUR POINTS HAVE BEEN DETERMINED
 -2.947 2.000
 -7.608 10.550

8.020	21.800
23.649	21.800
18.928	10.550
	2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 5.00

WAVELENGTH 10.60 MICROMETERS
 TRANSMITTER COORDINATES -200.00 .00 2.00
 RECEIVER COORDINATES 200.00 .00 2.00
 TRANSMITTANCE ALONG THE LINE OF SIGHT .940+000

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 23.50 .00
 THE HEIGHT OF THE CLOUD IS 23.50 METERS
 THE CENTROID COORDINATES ARE 14.70 10.50
 THE WIDTH AT THE CENTROID IS 38.28 METERS
 THE WIDTH AT 2.00 METERS IS 23.37 METERS
 6 CONTOUR POINTS HAVE BEEN DETERMINED

-2.24	2.000
-4.436	10.550
14.704	23.550
14.704	23.550
23.844	10.550
29.151	2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 7.00

WAVELENGTH 10.60 MICROMETERS
 TRANSMITTER COORDINATES -200.00 .00 2.00
 RECEIVER COORDINATES 200.00 .00 2.00
 TRANSMITTANCE ALONG THE LINE OF SIGHT .987+000

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 24.95 .00
 THE HEIGHT OF THE CLOUD IS 24.95 METERS
 THE CENTROID COORDINATES ARE 21.39 10.55
 THE WIDTH AT THE CENTROID IS 43.76 METERS
 THE WIDTH AT 2.00 METERS IS 35.94 METERS
 6 CONTOUR POINTS HAVE BEEN DETERMINED

3.436	2.000
-4.91	10.550
21.398	24.948
21.398	24.948
43.267	10.550
39.374	2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 9.00

WAVELENGTH 10.60 MICROMETERS
 TRANSMITTER COORDINATES -200.00 .00 2.00
 RECEIVER COORDINATES 200.00 .00 2.00
 TRANSMITTANCE ALONG THE LINE OF SIGHT .997+000

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 26.15 .00
 THE HEIGHT OF THE CLOUD IS 26.15 METERS
 THE CENTROID COORDINATES ARE 28.07 10.55
 THE WIDTH AT THE CENTROID IS 48.33 METERS
 THE WIDTH AT 2.00 METERS IS 41.56 METERS
 6 CONTOUR POINTS HAVE BEEN DETERMINED

7.097	2.000
-------	-------

WAVELENGTH	10.60	10.550
TRANSMITTER COORDINATES	-200.00	26.148
RECEIVER COORDINATES	200.00	26.148
TRANSMITTANCE ALONG THE LINE OF SIGHT	.999+000	10.550
		2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 11.00

WAVELENGTH	10.60 MICROMETERS	
TRANSMITTER COORDINATES	-200.00	.00 2.00
RECEIVER COORDINATES	200.00	.00 2.00
TRANSMITTANCE ALONG THE LINE OF SIGHT	.999+000	

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES	200.00	27.21 .00
THE HEIGHT OF THE CLOUD IS		34.78 10.55
THE CENTROID COORDINATES ARE		52.30 METERS
THE WIDTH AT THE CENTROID IS		46.56 METERS
THE WIDTH AT 2.00 METERS IS		
6 CONTOUR POINTS HAVE BEEN DETERMINED		
	11.38	2.000
	13.60	10.550
	34.78	27.21
	44.75	27.21
	60.904	10.550
	57.944	2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 13.00

WAVELENGTH	10.60 MICROMETERS	
TRANSMITTER COORDINATES	-200.00	.00 2.00
RECEIVER COORDINATES	200.00	.00 2.00
TRANSMITTANCE ALONG THE LINE OF SIGHT	.100+001	

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES	200.00	28.17 .00
THE HEIGHT OF THE CLOUD IS		41.44 10.55
THE CENTROID COORDINATES ARE		55.82 METERS
THE WIDTH AT THE CENTROID IS		50.94 METERS
THE WIDTH AT 2.00 METERS IS		
6 CONTOUR POINTS HAVE BEEN DETERMINED		
	15.86	2.000
	13.52	10.550
	41.43	28.17
	41.43	28.17
	69.34	10.550
	66.605	2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 15.00

WAVELENGTH	10.60 MICROMETERS	
TRANSMITTER COORDINATES	-200.00	.00 2.00
RECEIVER COORDINATES	200.00	.00 2.00
TRANSMITTANCE ALONG THE LINE OF SIGHT	.100+001	

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES	200.00	29.05 .00
THE HEIGHT OF THE CLOUD IS		48.12 10.55
THE CENTROID COORDINATES ARE		59.00 METERS
THE WIDTH AT THE CENTROID IS		55.00 METERS
THE WIDTH AT 2.00 METERS IS		
6 CONTOUR POINTS HAVE BEEN DETERMINED		

20.265	2.000
18.624	10.550
48.123	29.050
48.123	29.050
77.624	10.550
75.265	2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 17.00

WAVELENGTH	10.60 MICROMETERS	
TRANSMITTER COORDINATES	200.00	.00 2.00
RECEIVER COORDINATES	200.00	.00 2.00
TRANSMITTANCE ALONG THE LINE OF SIGHT	.100+001	

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES	200.00	29.86 .00
THE HEIGHT OF THE CLOUD IS		54.81 10.55
THE CENTROID COORDINATES ARE		61.91 METERS
THE WIDTH AT THE CENTROID IS		59.38 METERS
THE WIDTH AT 2.00 METERS IS		
6 CONTOUR POINTS HAVE BEEN DETERMINED		
	24.551	2.000
	53.850	10.550
	54.807	29.859
	54.807	29.859
	85.763	10.550
	83.926	2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 19.00

WAVELENGTH	10.60 MICROMETERS	
TRANSMITTER COORDINATES	200.00	.00 2.00
RECEIVER COORDINATES	200.00	.00 2.00
TRANSMITTANCE ALONG THE LINE OF SIGHT	.100+001	

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES	200.00	30.61 .00
THE HEIGHT OF THE CLOUD IS		61.49 10.55
THE CENTROID COORDINATES ARE		64.60 METERS
THE WIDTH AT THE CENTROID IS		63.12 METERS
THE WIDTH AT 2.00 METERS IS		
6 CONTOUR POINTS HAVE BEEN DETERMINED		
	29.148	2.000
	29.190	10.550
	61.490	30.611
	61.490	30.611
	93.791	10.550
	92.273	2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 21.00

WAVELENGTH	10.60 MICROMETERS	
TRANSMITTER COORDINATES	200.00	.00 2.00
RECEIVER COORDINATES	200.00	.00 2.00
TRANSMITTANCE ALONG THE LINE OF SIGHT	.100+001	

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES	200.00	31.32 .00
THE HEIGHT OF THE CLOUD IS		68.17 10.55
THE CENTROID COORDINATES ARE		67.10 METERS
THE WIDTH AT THE CENTROID IS		66.87 METERS
THE WIDTH AT 2.00 METERS IS		

6 CONTOUR POINTS HAVE BEEN DETERMINED

33.434	2.000
34.623	10.550
68.174	31.315
68.174	31.315
101.725	10.550
100.309	2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 23.00

WAVELENGTH	10.60 MICROMETERS	
TRANSMITTER COORDINATES	-200.00	.00
RECEIVER COORDINATES	200.00	.00
TRANSMITTANCE ALONG THE LINE OF SIGHT		.100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES	200.00	.00
THE HEIGHT OF THE CLOUD IS		31.98 METERS
THE CENTROID COORDINATES ARE		74.86 10.55
THE WIDTH AT THE CENTROID IS		69.44 METERS
THE WIDTH AT 2.00 METERS IS		70.31 METERS
6 CONTOUR POINTS HAVE BEEN DETERMINED		
	38.032	2.000
	40.135	10.550
	74.858	31.977
	74.858	31.977
	109.580	10.550
	108.344	2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 25.00

WAVELENGTH	10.60 MICROMETERS	
TRANSMITTER COORDINATES	-200.00	.00
RECEIVER COORDINATES	200.00	.00
TRANSMITTANCE ALONG THE LINE OF SIGHT		.100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES	200.00	.00
THE HEIGHT OF THE CLOUD IS		32.60 METERS
THE CENTROID COORDINATES ARE		81.54 10.55
THE WIDTH AT THE CENTROID IS		71.65 METERS
THE WIDTH AT 2.00 METERS IS		73.75 METERS
6 CONTOUR POINTS HAVE BEEN DETERMINED		
	42.630	2.000
	45.717	10.550
	81.541	32.603
	81.541	32.603
	117.366	10.550
	116.380	2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 27.00

WAVELENGTH	10.60 MICROMETERS	
TRANSMITTER COORDINATES	-200.00	.00
RECEIVER COORDINATES	200.00	.00
TRANSMITTANCE ALONG THE LINE OF SIGHT		.100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES	200.00	.00
THE HEIGHT OF THE CLOUD IS		33.20 METERS
THE CENTROID COORDINATES ARE		68.53 10.55
THE WIDTH AT THE CENTROID IS		73.73 METERS

THE WIDTH AT 2.00 METERS IS 77.19 METERS
 6 CONTOUR POINTS HAVE BEEN DETERMINED
 47.227 2.000
 51.553 10.550
 83.227 33.196
 83.227 33.196
 124.415 10.550
 124.415 2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 29.00

WAVELENGTH 10.60 MICROMETERS
 TRANSMITTER COORDINATES -200.00 .00 2.00
 RECEIVER COORDINATES 200.00 .00 2.00
 TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
 THE HEIGHT OF THE CLOUD IS 33.76 METERS
 THE CENTROID COORDINATES ARE 94.91 10.55
 THE WIDTH AT THE CENTROID IS 75.71 METERS
 THE WIDTH AT 2.00 METERS IS 80.94 METERS
 6 CONTOUR POINTS HAVE BEEN DETERMINED
 51.513 2.000
 57.055 10.550
 94.909 33.761
 94.909 33.761
 132.763 10.550
 132.450 2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 31.00

WAVELENGTH 10.60 MICROMETERS
 TRANSMITTER COORDINATES -200.00 .00 2.00
 RECEIVER COORDINATES 200.00 .00 2.00
 TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
 THE HEIGHT OF THE CLOUD IS 34.30 METERS
 THE CENTROID COORDINATES ARE 101.59 10.55
 THE WIDTH AT THE CENTROID IS 77.59 METERS
 THE WIDTH AT 2.00 METERS IS 84.06 METERS
 6 CONTOUR POINTS HAVE BEEN DETERMINED
 56.111 2.000
 62.799 10.550
 101.593 34.299
 101.593 34.299
 140.387 10.550
 140.173 2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 33.00

WAVELENGTH 10.60 MICROMETERS
 TRANSMITTER COORDINATES -200.00 .00 2.00
 RECEIVER COORDINATES 200.00 .00 2.00
 TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
 THE HEIGHT OF THE CLOUD IS 34.81 METERS
 THE CENTROID COORDINATES ARE 108.28 10.55

THE WIDTH AT THE CENTROID IS 79.38 METERS
 THE WIDTH AT 2.00 METERS IS 87.19 METERS
 6 CONTOUR POINTS HAVE BEEN DETERMINED

80.700	2.000
98.550	10.550
108.276	34.814
108.276	34.814
147.960	10.550
147.896	2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 35.00

WAVELENGTH 10.60 MICROMETERS
 TRANSMITTER COORDINATES -200.00 .00 2.00
 RECEIVER COORDINATES 200.00 .00 2.00
 TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
 THE HEIGHT OF THE CLOUD IS 35.31 METERS
 THE CENTROID COORDINATES ARE 114.96 10.55
 THE WIDTH AT THE CENTROID IS 81.10 METERS
 THE WIDTH AT 2.00 METERS IS 90.31 METERS
 6 CONTOUR POINTS HAVE BEEN DETERMINED

65.306	2.000
74.410	10.550
114.960	35.308
114.960	35.308
155.510	10.550
155.619	2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 37.00

WAVELENGTH 10.60 MICROMETERS
 TRANSMITTER COORDINATES -200.00 .00 2.00
 RECEIVER COORDINATES 200.00 .00 2.00
 TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
 THE HEIGHT OF THE CLOUD IS 35.78 METERS
 THE CENTROID COORDINATES ARE 121.64 10.55
 THE WIDTH AT THE CENTROID IS 82.75 METERS
 THE WIDTH AT 2.00 METERS IS 93.75 METERS
 6 CONTOUR POINTS HAVE BEEN DETERMINED

69.592	2.000
80.271	10.550
121.644	35.782
121.644	35.782
163.017	10.550
163.342	2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 39.00

WAVELENGTH 10.60 MICROMETERS
 TRANSMITTER COORDINATES -200.00 .00 2.00
 RECEIVER COORDINATES 200.00 .00 2.00
 TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
 THE HEIGHT OF THE CLOUD IS 36.24 METERS

THE CENTROID COORDINATES ARE 128.35 10.55
 THE WIDTH AT THE CENTROID IS 84.33 METERS
 THE WIDTH AT 2.00 METERS IS 96.88 METERS
 6 CONTOUR POINTS HAVE BEEN DETERMINED

74.190	2.000
86.164	10.550
128.328	36.638
128.328	36.638
170.492	10.550
171.065	2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 41.00

WAVELENGTH 10.60 MICROMETERS
 TRANSMITTER COORDINATES -200.00 .00 2.00
 RECEIVER COORDINATES 200.00 .00 2.00
 TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
 THE HEIGHT OF THE CLOUD IS 36.68 METERS
 THE CENTROID COORDINATES ARE 135.01 10.55
 THE WIDTH AT THE CENTROID IS 85.85 METERS
 THE WIDTH AT 2.00 METERS IS 100.00 METERS
 6 CONTOUR POINTS HAVE BEEN DETERMINED

78.787	2.000
92.086	10.550
135.011	36.677
135.011	36.677
177.936	10.550
178.787	2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 43.00

WAVELENGTH 10.60 MICROMETERS
 TRANSMITTER COORDINATES -200.00 .00 2.00
 RECEIVER COORDINATES 200.00 .00 2.00
 TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
 THE HEIGHT OF THE CLOUD IS 37.10 METERS
 THE CENTROID COORDINATES ARE 141.70 10.55
 THE WIDTH AT THE CENTROID IS 87.32 METERS
 THE WIDTH AT 2.00 METERS IS 102.81 METERS
 6 CONTOUR POINTS HAVE BEEN DETERMINED

83.385	2.000
98.037	10.550
141.695	37.102
141.695	37.102
185.353	10.550
186.198	2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 45.00

WAVELENGTH 10.60 MICROMETERS
 TRANSMITTER COORDINATES -200.00 .00 2.00
 RECEIVER COORDINATES 200.00 .00 2.00
 TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00

THE HEIGHT OF THE CLOUD IS 37.51 METERS
 THE CENTROID COORDINATES ARE 148.38 10.55
 THE WIDTH AT THE CENTROID IS 88.73 METERS
 THE WIDTH AT 2.00 METERS IS 105.94 METERS
 6 CONTOUR POINTS HAVE BEEN DETERMINED
 87.983 2.000
 104.013 10.550
 148.379 37.512
 148.379 37.512
 132.745 10.550
 133.921 2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 47.00

WAVELENGTH 10.60 MICROMETERS
 TRANSMITTER COORDINATES -200.00 .00 2.00
 RECEIVER COORDINATES 200.00 .00 2.00
 TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 200.00
 THE HEIGHT OF THE CLOUD IS 37.91 METERS
 THE CENTROID COORDINATES ARE 155.06 10.55
 THE WIDTH AT THE CENTROID IS 90.10 METERS
 THE WIDTH AT 2.00 METERS IS 108.75 METERS
 6 CONTOUR POINTS HAVE BEEN DETERMINED
 92.581 2.000
 110.012 10.550
 155.063 37.909
 155.063 37.909
 200.113 10.550
 201.331 2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 49.00

WAVELENGTH 10.60 MICROMETERS
 TRANSMITTER COORDINATES -200.00 .00 2.00
 RECEIVER COORDINATES 200.00 .00 2.00
 TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 200.00
 THE HEIGHT OF THE CLOUD IS 38.29 METERS
 THE CENTROID COORDINATES ARE 161.75 10.55
 THE WIDTH AT THE CENTROID IS 91.43 METERS
 THE WIDTH AT 2.00 METERS IS 111.87 METERS
 6 CONTOUR POINTS HAVE BEEN DETERMINED
 97.175 2.000
 116.034 10.550
 161.746 38.293
 161.746 38.293
 207.459 10.550
 209.054 2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 51.00

WAVELENGTH 10.60 MICROMETERS
 TRANSMITTER COORDINATES -200.00 .00 2.00
 RECEIVER COORDINATES 200.00 .00 2.00
 TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 38.67 .00
 THE HEIGHT OF THE CLOUD IS 38.67 METERS
 THE CENTROID COORDINATES ARE 168.43 10.55
 THE WIDTH AT THE CENTROID IS 92.71 METERS
 THE WIDTH AT 2.00 METERS IS 114.69 METERS
 6 CONTOUR POINTS HAVE BEEN DETERMINED
 101.777 2.000
 122.076 10.550
 168.430 38.665
 168.430 38.665
 214.784 10.550
 216.464 2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 53.00

WAVELENGTH 10.60 MICROMETERS
 TRANSMITTER COORDINATES -200.00 .00 2.00
 RECEIVER COORDINATES 200.00 .00 2.00
 TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 39.03 .00
 THE HEIGHT OF THE CLOUD IS 39.03 METERS
 THE CENTROID COORDINATES ARE 175.11 10.55
 THE WIDTH AT THE CENTROID IS 93.95 METERS
 THE WIDTH AT 2.00 METERS IS 117.50 METERS
 6 CONTOUR POINTS HAVE BEEN DETERMINED
 106.375 2.000
 128.138 10.550
 175.114 39.027
 175.114 39.027
 222.090 10.550
 223.875 2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 55.00

WAVELENGTH 10.60 MICROMETERS
 TRANSMITTER COORDINATES -200.00 .00 2.00
 RECEIVER COORDINATES 200.00 .00 2.00
 TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 39.38 .00
 THE HEIGHT OF THE CLOUD IS 39.38 METERS
 THE CENTROID COORDINATES ARE 181.80 10.55
 THE WIDTH AT THE CENTROID IS 95.16 METERS
 THE WIDTH AT 2.00 METERS IS 120.00 METERS
 6 CONTOUR POINTS HAVE BEEN DETERMINED
 111.285 2.000
 134.217 10.550
 181.797 39.378
 181.797 39.378
 229.377 10.550
 231.285 2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 57.00

WAVELENGTH 10.60 MICROMETERS
 TRANSMITTER COORDINATES -200.00 .00 2.00
 RECEIVER COORDINATES 200.00 .00 2.00
 TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00
 THE HEIGHT OF THE CLOUD IS 39.72 METERS
 THE CENTROID COORDINATES ARE 188.48 10.55
 THE WIDTH AT THE CENTROID IS 96.33 METERS
 THE WIDTH AT 2.00 METERS IS 122.81 METERS
 6 CONTOUR POINTS HAVE BEEN DETERMINED
 115.883 2.000
 140.315 10.550
 188.481 39.720
 188.481 39.720
 238.648 10.550
 238.695 2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 59.00

WAVELENGTH 10.60 MICROMETERS
 TRANSMITTER COORDINATES -200.00 .00 2.00
 RECEIVER COORDINATES 200.00 .00 2.00
 TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00
 THE HEIGHT OF THE CLOUD IS 40.05 METERS
 THE CENTROID COORDINATES ARE 195.16 10.55
 THE WIDTH AT THE CENTROID IS 97.47 METERS
 THE WIDTH AT 2.00 METERS IS 125.62 METERS
 6 CONTOUR POINTS HAVE BEEN DETERMINED
 120.481 2.000
 146.428 10.550
 195.165 40.052
 195.165 40.052
 243.902 10.550
 246.106 2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 61.00

WAVELENGTH 10.60 MICROMETERS
 TRANSMITTER COORDINATES -200.00 .00 2.00
 RECEIVER COORDINATES 200.00 .00 2.00
 TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00
 THE HEIGHT OF THE CLOUD IS 40.38 METERS
 THE CENTROID COORDINATES ARE 201.85 10.55
 THE WIDTH AT THE CENTROID IS 98.58 METERS
 THE WIDTH AT 2.00 METERS IS 128.12 METERS
 6 CONTOUR POINTS HAVE BEEN DETERMINED
 125.391 2.000
 152.557 10.550
 201.849 40.376
 201.849 40.376
 251.140 10.550
 253.516 2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 63.00

WAVELENGTH 10.60 MICROMETERS
 TRANSMITTER COORDINATES -200.00 .00 2.00
 RECEIVER COORDINATES 200.00 .00 2.00
 TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 40.69 .00
 THE HEIGHT OF THE CLOUD IS 208.53 10.55
 THE CENTROID COORDINATES ARE 99.66 METERS
 THE WIDTH AT THE CENTROID IS 130.62 METERS
 THE WIDTH AT 2.00 METERS IS
 6 CONTOUR POINTS HAVE BEEN DETERMINED
 130.301 2.000
 156.700 10.550
 208.532 40.691
 208.532 40.691
 258.364 10.550
 260.926 2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 65.00

WAVELENGTH 10.60 MICROMETERS
 TRANSMITTER COORDINATES -200.00 .00 2.00
 RECEIVER COORDINATES 200.00 .00 2.00
 TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 41.00 .00
 THE HEIGHT OF THE CLOUD IS 215.22 10.55
 THE CENTROID COORDINATES ARE 100.72 METERS
 THE WIDTH AT THE CENTROID IS 133.12 METERS
 THE WIDTH AT 2.00 METERS IS
 6 CONTOUR POINTS HAVE BEEN DETERMINED
 135.212 2.000
 164.857 10.550
 215.216 40.998
 215.216 40.998
 265.575 10.550
 268.337 2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 67.00

WAVELENGTH 10.60 MICROMETERS
 TRANSMITTER COORDINATES -200.00 .00 2.00
 RECEIVER COORDINATES 200.00 .00 2.00
 TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 41.30 .00
 THE HEIGHT OF THE CLOUD IS 221.90 10.55
 THE CENTROID COORDINATES ARE 101.74 METERS
 THE WIDTH AT THE CENTROID IS 135.00 METERS
 THE WIDTH AT 2.00 METERS IS
 6 CONTOUR POINTS HAVE BEEN DETERMINED
 140.435 2.000
 171.028 10.550
 221.900 41.298
 221.900 41.298
 272.772 10.550
 275.435 2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 69.00

WAVELENGTH 10.60 MICROMETERS
 TRANSMITTER COORDINATES -200.00 .00 2.00
 RECEIVER COORDINATES 200.00 .00 2.00
 TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 0.00
 THE HEIGHT OF THE CLOUD IS 41.59 METERS
 THE CENTROID COORDINATES ARE 228.58 10.55
 THE WIDTH AT THE CENTROID IS 103.75 METERS
 THE WIDTH AT 2.00 METERS IS 137.50 METERS
 6 CONTOUR POINTS HAVE BEEN DETERMINED
 145.345 2.000
 177.211 10.550
 228.584 41.591
 228.584 41.591
 279.956 10.550
 282.845 2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 71.00

WAVELENGTH 10.60 MICROMETERS
 TRANSMITTER COORDINATES -200.00 0.00 2.00
 RECEIVER COORDINATES 200.00 0.00 2.00
 TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 0.00
 THE HEIGHT OF THE CLOUD IS 41.88 METERS
 THE CENTROID COORDINATES ARE 235.27 10.55
 THE WIDTH AT THE CENTROID IS 103.72 METERS
 THE WIDTH AT 2.00 METERS IS 139.69 METERS
 6 CONTOUR POINTS HAVE BEEN DETERMINED
 150.568 2.000
 183.406 10.550
 235.267 41.877
 235.267 41.877
 287.129 10.550
 290.255 2.000

LASER TRANSMITTANCE MODULE

WAVELENGTH (MICRONS)	H2O PRESSURE (TORR)	TEMPERATURE (KES)	ABSORPTION COEFFICIENT (KM-1)	LINE	PATHLENGTH (KM)	TRANSMISSION
10.591	5.821	278.46	.111+000	P(20)	.4000+000	.9566+000

*** WARNING INPUT WAVELENGTH 10.600 CHANGED TO 10.591 NEAREST STANDARD WAVELENGTH ***

CLOUD TRANSMITTANCE MODULE

WAVELENGTH = 10.60 MICRONS

LINE-OF-SIGHT SLANTS DOWNWARD

TOTAL LINE-OF-SIGHT LENGTH = .719 KM
 TOTAL LINE-OF-SIGHT LENGTH INTERRUPTED BY CLOUD = .481 KM

TOTAL OPTICAL DEPTH = 25.11

TRANSMITTANCE ALONG LINE-OF-SIGHT = .12449-010

SEEKER COORDINATES (KM) TARGET COORDINATES (KM)
 XSEEKER YSEEKER ZSEEKER XTARGET YTARGET ZTARGET
 .200 .000 .600 -.200 .000 .002

CLOUD TYPE /ID NUMBER	LINE-OF-SIGHT		INTERSECTION		COORDINATES (KM)	
	XUPPER	YUPPER	ZUPPER	XLOWER	YLOWER	ZLOWER
ST/ 1	.200	.000	.600	-.068	.000	.200

CLOUD TYPE /ID NUMBER	HEIGHT OF BASE (KM)	THICKNESS (KM)	RADIUS OF CLOUD (KM)	OPTICAL DEPTH		TRANSMITTANCE ALONG L-O-S
				ALONG L-O-S	ALONG L-O-S	
ST/ 1	.200	.400	.000	25.11		.12449-010

CLOUD TRANSMITTANCE MODULE

WAVELENGTH = 10.60 MICRONS

LINE-OF-SIGHT SLANTS DOWNWARD

TOTAL LINE-OF-SIGHT LENGTH = .599 KM
 TOTAL LINE-OF-SIGHT LENGTH INTERRUPTED BY CLOUD = .361 KM

TOTAL OPTICAL DEPTH = 16.57

TRANSMITTANCE ALONG LINE-OF-SIGHT = .63594-007

SEEKER COORDINATES (KM)	TARGET COORDINATES (KM)
XSEEKER YSEEKER ZSEEKER	XTARGET YTARGET ZTARGET
.133 .000 .500	-.200 .000 .002

CLOUD TYPE /ID NUMBER	HEIGHT OF BASE (KM)	THICKNESS (KM)	RADIUS OF CLOUD (KM)	OPTICAL DEPTH ALONG L-O-S	TRANSMITTANCE ALONG L-O-S
ST/ 1	.200	.400	.000	16.57	.63594-007

CLOUD TRANSMITTANCE MODULE

WAVELENGTH = 10.60 MICRONS

LINE-OF-SIGHT SLANTS DOWNWARD

TOTAL LINE-OF-SIGHT LENGTH = .359 KM
 TOTAL LINE-OF-SIGHT LENGTH INTERRUPTED BY CLOUD = .120 KM

TOTAL OPTICAL DEPTH = 3.02

TRANSMITTANCE ALONG LINE-OF-SIGHT = .48846-001

SEEKER COORDINATES (KM) TARGET COORDINATES (KM)
 XSEEKER YSEEKER ZSEEKER XTARGET YTARGET ZTARGET

 .000 .000 .300 -.200 .000 .002

CLOUD TYPE /ID NUMBER	LINE-OF-SIGHT		INTERSECTION		COORDINATES (KM)	
	XUPPER	YUPPER	ZUPPER	XLOWER	YLOWER	ZLOWER
ST/ 1	.000	.000	.300	-.067	.000	.200

CLOUD TYPE /ID NUMBER	HEIGHT OF BASE (KM)	THICKNESS (KM)	RADIUS OF CLOUD (KM)	OPTICAL DEPTH ALONG L-O-S	TRANSMITTANCE ALONG L-O-S
ST/ 1	.200	.400	.000	3.02	.48846-001

CWIC MUNITION EXPENDITURES / INVERSE STATIC TARGET DETECTION MODULE

 ** INVERSE STATIC TARGET DETECTION MODEL **

DEVICE NUMBER	8	TARGET INTRINSIC CONTRAST OR TEMPERATURE DIFFERENCE	1.000	COMMENTS
FOV TYPE -	WIDE	MINIMUM TARGET DIMENSION (M)	2.300	
FOV (DEG)	10.620	ACQUISITION LEVEL (50 PCNT)	1.000	
MAGNIFICATION	1.000 (ARBITRARY DEFAULT)	SEARCH ZONE (DEGREES**2)	81.000	
FOR NO GREATER DETECTION PROB. TIME(SEC)	1.32	REQUIRES (TO DEFEAT DEVICE) AT MOST COMPUTED CONTRAST OR COMPUTED RESOLVABLE TEMP. DIFF. TOTAL PATH CYCLES, RC AT DEVICE. TRANSMITTANCE	.037	NOTE- INPUT DETECTION PROBAB- ILITY REQUIRES CONTRAST (OR TEMP. DIFF.) BELOW THRESHOLD. VALUES ASSUMED ARE 99 PERCENT OF THRESHOLD. ADDITIONAL OBS- CURANT WILL NOT DECREASE DETECTION PROBABILITY.
	.990		.037	
	1.59		.037	NOTE- INPUT DETECTION PROBAB- ILITY REQUIRES CONTRAST (OR TEMP. DIFF.) BELOW THRESHOLD. VALUES ASSUMED ARE 99 PERCENT OF THRESHOLD. ADDITIONAL OBS- CURANT WILL NOT DECREASE DETECTION PROBABILITY.
	.900		.037	
	2.16		.037	NOTE- INPUT DETECTION PROBAB- ILITY REQUIRES CONTRAST (OR TEMP. DIFF.) BELOW THRESHOLD. VALUES ASSUMED ARE 99 PERCENT OF THRESHOLD. ADDITIONAL OBS- CURANT WILL NOT DECREASE DETECTION PROBABILITY.
	.750		.037	
	3.89		.037	NOTE- INPUT DETECTION PROBAB- ILITY REQUIRES CONTRAST (OR TEMP. DIFF.) BELOW THRESHOLD. VALUES ASSUMED ARE 99 PERCENT OF THRESHOLD. ADDITIONAL OBS- CURANT WILL NOT DECREASE DETECTION PROBABILITY.
	.500		.037	

** CONTINUED ON NEXT PAGE **

 ** INVERSE STATIC TARGET DETECTION MODEL **

DEVICE NUMBER 8
 FOV TYPE - WIDE
 FOV (DEG) 10.620
 MAGNIFICATION 1.000 (ARBITRARY DEFAULT)

TARGET INTRINSIC CONTRAST OR
 TEMPERATURE DIFFERENCE

1.000

MINIMUM TARGET DIMENSION (M)

2.300

ACQUISITION LEVEL (50 PCNT)

1.000

SEARCH ZONE (DEGREES**2)

81.000

REQUIRES (TO DEFEAT DEVICE) AT MOST

COMPUTED CONTRAST OR

RESOLVABLE TEMP. DIFF. TOTAL PATH

CYCLES, RC AT DEVICE TRANSMITTANCE

.037

.037

.059

.24

2.000

10.000000

FOR NO GREATER

INPUT

DETECTION

PROB. TIME (SEC)

.100 24.64

COMMENTS

NOTE - INPUT DETECTION PROB-
 ILITY REQUIRES CONTRAST (OR
 TEMP. DIFF.) BELOW PRESENT
 VALUES ASSUMED. RARE PRESENT
 OF THIS THRESHOLD. NO DETECTION
 OF TARGET WILL OCCUR. INCREASE
 DETECTION PROBABILITY.

*** FINAL TOTAL TRANSMISSION FROM ITAM = .037

 * * * * *
 * * * * *
 * * * * *
 * * * * *
 * * * * *

MUNITION EXPENDITURES
 FOR HC AND WP SMOKE

SLANT RANGE OBS-TGT	- KM	=	.400
ELEVATION OF TARGET	- DEG	=	.00
AZIMUTH OF TARGET	- DEG	=	90.00
AVG. ROUGHNESS ELEMENT	- CM	=	74.0
ATMOSPHERIC EXTINCTION CORRECTIONS			
CORRECTED FOR VISIBILITY	-		YES
MARITIME ARCTIC AIR MASS	-		YES
MARITIME POLAR AIR MASS	-		NO
CONTINENTAL POLAR AIR MASS	-		NO
CORRECTED FOR RAIN	-		NO
CORRECTED FOR SNOW	-		NO
METEOROLOGICAL INPUTS			
WINDSPEED	- M/SEC	=	2.60
WIND DIRECTION	- DEG	=	225.00
WIND CATEGORY	-	=	0
WIND QUALITY	- KM	=	9.302
RELATIVE HUMIDITY	- PERCENT	=	87.1
TRANSMISSION THRESHOLDS			
TOTAL			SMOKE
VISIBILITY			.037
NEAR IR			.043
MID IR			.041
FAR IR			.039

----- VISIBLE: -----		----- NEAR IR: -----	
SCREEN	LENGTH METERS 400.	SCREEN	LENGTH METERS 400.
	DURATION MINUTES 15.00		DURATION MINUTES 15.00
HC SMOKE SCREEN			
105MM HOWITZER			
VOLLEY	GUNS RATE /MIN 8.	VOLLEY	GUNS RATE /MIN 23.
INITIAL:	51.	INITIAL:	18.
SUSTAINING:	51.	SUSTAINING:	18.
155MM HOWITZER			
VOLLEY	GUNS RATE /MIN 3.	VOLLEY	GUNS RATE /MIN 4.
INITIAL:	162.	INITIAL:	114.
SUSTAINING:	328.	SUSTAINING:	114.
UP SMOKE SCREEN			
105MM HOWITZER			
VOLLEY	GUNS RATE /MIN 6.	VOLLEY	GUNS RATE /MIN 8.
INITIAL:	82.	INITIAL:	66.
SUSTAINING:	129.	SUSTAINING:	172.
155MM HOWITZER			
VOLLEY	GUNS RATE /MIN 3.	VOLLEY	GUNS RATE /MIN 4.
INITIAL:	246.	INITIAL:	186.
SUSTAINING:	31.	SUSTAINING:	41.

- MID IR: -
 LENGTH METERS 400.
 DURATION MINUTES 15.00
 SCREEN
 UP SMOKE SCREEN
 ROUNDS/ RATE/ TOTAL
 60 METERS MINUTE ROUNDS
 105MM: 3. 1. 435.
 155MM: 3. 1. 435.

- FAR IR: -
 LENGTH METERS 400.
 DURATION MINUTES 15.00
 SCREEN
 UP SMOKE SCREEN
 ROUNDS/ RATE/ TOTAL
 60 METERS MINUTE ROUNDS
 105MM: 2. 1. 330.
 155MM: 2. 1. 330.

OVERCAST SKY RADIATIVE TRANSFER MODULE

```
-- RADIATION UNDER OVERCAST SKY --
X0 = -0.067 (KM)      XT = -200 (KM)
X1 = -200 (KM)         Y1 = 0.000
X2 = -200 (KM)         Y2 = 0.000
ZC = 10.200 (MU)       LC = 2.0000+002 (W/M2-SR-MU)
LAMBDA = 10.200 (MU)   LG = 5.0000+001
TEMP = 9.8 (DEG.C)     LB0 = 5.0000+001
KAPPA = 4.0000+001 (KM-1) W0 = 1.000
ETA = .700
```

**THERMAL CALCULATION OF PATH RADIANCE

BBTEMP= 7.4082+000 W/M2-SR-MU

```
PATH LENGTH (KM)  TRANSMITTANCE  PATH RADIANCE  CONTRAST
+-----+-----+-----+-----+
.239             .00007      2.409+001      .00015
```

```
-- RADIATION UNDER OVERCAST SKY --
X0 = -.133 (KM)      XT = -200 (KM)
X1 = -200 (KM)         Y1 = 0.000
X2 = -200 (KM)         Y2 = 0.000
ZC = 10.200 (MU)       LC = 2.0000+002 (W/M2-SR-MU)
LAMBDA = 10.200 (MU)   LG = 5.0000+001
TEMP = 9.8 (DEG.C)     LB0 = 5.0000+001
KAPPA = 4.0000+001 (KM-1) W0 = 1.000
ETA = .700
```

**THERMAL CALCULATION OF PATH RADIANCE

BBTEMP= 7.4082+000 W/M2-SR-MU

```
PATH LENGTH (KM)  TRANSMITTANCE  PATH RADIANCE  CONTRAST
+-----+-----+-----+-----+
.119             .00872      9.826+001      .30742
```

SELF-SCREENING SMOKE GRENADE MODULE

 * PROGRAM CENADE *****
 * EUSAE 80 *****
 * *****

*****CARD INPUT*****

NAME	OUTP	.000	.000	.000	.000	.000	.000	.000	.000
OBSC	200.000	.000	.000	2.000	.000	.000	.000	.000	.000
MUNC	-200.000	.000	.000	2.000	95.000	100.000	10.000	.000	.000
TARC	-200.000	40.000	.000	2.000	.000	.000	.000	.000	.000
BART	5.000	400.000	.000	5.000	90.000	.000	.000	.000	.000
MUNT	1.000	.793	14.300	1.000	.000	4.700	.070	.000	.000
METR	50.000	2.000	220.000	4.000	20.000	.000	.000	.000	.000
EXTC	.000	.000	.000	.000	.000	.000	.000	.000	.000
BURN	.000	.000	.000	.000	.000	.000	.000	.000	.000
MISC	.000	.000	.000	.000	.000	.000	.000	.000	.000
GO	.000	.000	.000	.000	.000	.000	.000	.000	.000

*****INPUT*****ALL LENGTHS IN METERS

METEOROLOGICAL:
 WIND SPEED 3.6 M/S
 WIND DIRECTION 225.0 DEC
 PASQUILL CATEGORY 4
 RELATIVE HUMIDITY 87.1 %
 NOTE: X AXIS HEADING: EXTINCTION COEFFICIENTS: 90.0 DEG CLOCKWISE FROM NORTH (DCWFN)
 OBSERVER/TARGET: MICRONS M**2/GM
 X(OBS) 200.0 0.7-1.2 4.304
 Y(OBS) 2.0 1.566
 X(TAR) -200.0 1.551
 Y(TAR) 40.0 .379
 Z(TAR) 2.0 .001
 94.0 CHZ DIFFUSION PARAMETERS:
 SIGZ(XREF)
 XREF
 MIXING HEIGHT(MH) 7.2
 SCAVENGING COEFF(HK) 100.0
 REFLECTION COEFF(RC) 300.0
 SETTLING VELOCITY(VS) .021 CM/S
 VERTICAL WIND EXPONENT(WPOMR) .000
 VERTICAL DIFF CONSTANT(ZDIFF) 1.400
 CROSSWIND DIFF CONSTANT(YDIFF) 1.355
 YIELD FACTOR 4.7

TANK/MUNITION DATA:
 X(TANK) -200.0
 Y(TANK) 2.0
 Z(TANK) 2.0
 HEADING(DCWFN) 95.0
 RANGE 100.0
 NO GRENADES 359.7
 SMOKE MASS(GM) 10.0
 LINE LENGTH .070 1/S
 BURN CONSTANT 62.0
 EFFICIENCY

OUTPUT

385.0	.277	.204	.549	.653	.308	.311	.304	1.000
390.0	.327	.259	.544	.627	.309	.312	.305	1.000
395.0	.370	.292	.539	.620	.310	.313	.302	1.000
400.0	.466	.358	.532	.664	.311	.314	.308	1.000

*****CARD INPUT*****

DONE .000

*****PROGRAM CRNADE END***** .000 .000 .000 .000

COMBINED TRANSMISSION FOR THE SELECTED MODULES = .2331-019

RUN NUMBER 2

INDIVIDUAL MODULES SELECTED
BASCAT
PCL000

	BEGINNING	ENDING
WAVENUMBER(CH**1)	9433.962	9433.962
WAVELENGTH(MICRONS)	1.060	1.060
FREQUENCY(GHZ)	283018.863	283018.863

CLIMATOLOGY MODEL

DEFINITIONS OF METEOROLOGICAL CLASSES

```

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CLAS FOG, HAZE AND MIST WITH VIS LT 1 KM. 3 KM.
CLAS FOG, HAZE AND MIST WITH 1 LE VIS LT 3 KM.
CLAS FOG, HAZE AND MIST WITH 3 LE VIS LT 7 KM.
CLAS FOG, HAZE AND MIST WITH VIS GE 7 KM.
CLAS DUST WITH VIS LT 1 KM.
CLAS DRIZZLE, RAIN AND TSMS WITH VIS LT 1 KM.
CLAS DRIZZLE, RAIN AND TSMS WITH 1 LE VIS LT 3 KM.
CLAS DRIZZLE, RAIN AND TSMS WITH 3 LE VIS LT 7 KM.
CLAS DRIZZLE, RAIN AND TSMS WITH VIS GE 7 KM.
CLAS SNOW WITH 1 LE VIS LT 3 KM.
CLAS SNOW WITH 3 LE VIS LT 7 KM.
CLAS SNOW WITH VIS GE 7 KM.
CLAS NO WEATHER AND ABSOLUTE HUMIDITY LT 10 GM/CM M.
CLAS NO WEATHER AND ABSOLUTE HUMIDITY GE 10 GM/CM M.
CLAS VIS LT 1 KM AND CEILING HEIGHT LT 300 M.
CLAS VIS LT 3 KM AND CEILING HEIGHT LT 1000 M.
CLAS CEILING HEIGHT LT 300 M.
CLAS CEILING HEIGHT LT 1000 M.
CLAS ALL CONDITIONS COMBINED.

```

ESRAEL CLIMATOLOGY FOR EUROPEAN LOWLANDS														DURING SPRING AT 03-09 (LST).	
CLASS NO.	FREQCY CLASS (%)	MEAN TEMP (C)	MEAN DEW (C)	MEAN RH (%)	MEAN WIND (MPS)	MEAN PRESS (MB)	MEAN WINDVEL (MPS)	MEAN CLOUDT (KM)	MEAN/STDEV CLOUDVR (%)	FREQCY A (%)	FREQCY B (%)	FREQCY C (%)	FREQCY D (%)		
1	0-10	15.0	10.0	65	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
2	10-20	16.0	11.0	68	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
3	20-30	17.0	12.0	70	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
4	30-40	18.0	13.0	72	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
5	40-50	19.0	14.0	75	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
6	50-60	20.0	15.0	78	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
7	60-70	21.0	16.0	80	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
8	70-80	22.0	17.0	82	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
9	80-90	23.0	18.0	85	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
10	90-100	24.0	19.0	88	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
11	100-110	25.0	20.0	90	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
12	110-120	26.0	21.0	92	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
13	120-130	27.0	22.0	95	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
14	130-140	28.0	23.0	98	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
15	140-150	29.0	24.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
16	150-160	30.0	25.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
17	160-170	31.0	26.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
18	170-180	32.0	27.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
19	180-190	33.0	28.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
20	190-200	34.0	29.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
21	200-210	35.0	30.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
22	210-220	36.0	31.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
23	220-230	37.0	32.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
24	230-240	38.0	33.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
25	240-250	39.0	34.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
26	250-260	40.0	35.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
27	260-270	41.0	36.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
28	270-280	42.0	37.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
29	280-290	43.0	38.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
30	290-300	44.0	39.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
31	300-310	45.0	40.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
32	310-320	46.0	41.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
33	320-330	47.0	42.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
34	330-340	48.0	43.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
35	340-350	49.0	44.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
36	350-360	50.0	45.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
37	360-370	51.0	46.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
38	370-380	52.0	47.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
39	380-390	53.0	48.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
40	390-400	54.0	49.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
41	400-410	55.0	50.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
42	410-420	56.0	51.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
43	420-430	57.0	52.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
44	430-440	58.0	53.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
45	440-450	59.0	54.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
46	450-460	60.0	55.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
47	460-470	61.0	56.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
48	470-480	62.0	57.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
49	480-490	63.0	58.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
50	490-500	64.0	59.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
51	500-510	65.0	60.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
52	510-520	66.0	61.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
53	520-530	67.0	62.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
54	530-540	68.0	63.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
55	540-550	69.0	64.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
56	550-560	70.0	65.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
57	560-570	71.0	66.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
58	570-580	72.0	67.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
59	580-590	73.0	68.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
60	590-600	74.0	69.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
61	600-610	75.0	70.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
62	610-620	76.0	71.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
63	620-630	77.0	72.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
64	630-640	78.0	73.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
65	640-650	79.0	74.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
66	650-660	80.0	75.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
67	660-670	81.0	76.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
68	670-680	82.0	77.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
69	680-690	83.0	78.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
70	690-700	84.0	79.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
71	700-710	85.0	80.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
72	710-720	86.0	81.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
73	720-730	87.0	82.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
74	730-740	88.0	83.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
75	740-750	89.0	84.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
76	750-760	90.0	85.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
77	760-770	91.0	86.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
78	770-780	92.0	87.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
79	780-790	93.0	88.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
80	790-800	94.0	89.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
81	800-810	95.0	90.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
82	810-820	96.0	91.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
83	820-830	97.0	92.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
84	830-840	98.0	93.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
85	840-850	99.0	94.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
86	850-860	100.0	95.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
87	860-870	101.0	96.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
88	870-880	102.0	97.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
89	880-890	103.0	98.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		
90	890-900	104.0	99.0	100	1.5	1010	1.5	0.5	10	0.0	0.0	0.0	0.0		

BASCAT LASER SCATTERING MODULE

 ** MONTE CARLO MULTIPLE SCATTERING **
 ** AEROSOL SCATTERING **

PARAMETERS FOR THIS RUN

WHITE PHOSPHORUS
 WAVELENGTH= 1.060 MICROMETERS ALBEDO= .999
 AEROSOL EXTINCTION COEFFICIENT= .4000+001 KM**-1
 ELLIPSOIDAL AEROSOL CLOUD
 COORDINATE ORIGIN AT CENTER OF CLOUD
 Z-AXIS VERTICAL, X-AXIS EAST, Y-AXIS NORTH
 SOURCE PARAMETERS
 SOURCE XYZ COORDINATES(KM)= -.2000 .0000 -.0980
 SOURCE AXIS POLAR ANGLE= 90.000 DEGREES
 SOURCE AXIS AZIMUTH ANGLE= 0.000 DEGREES
 SOURCE APERTURE RADIUS(MM)= 50.000
 SOURCE BEAM SPREAD ANGLE= .1293-004 RADIAN
 DETECTOR PARAMETERS
 CONE OF VIEW HALF-ANGLE= 1.000 DEGREES
 DETECTOR APERTURE RADIUS= 1.000 CM
 DETECTOR XYZ COORDINATES(KM)= .2000 .0000 -.0980
 DETECTOR AXIS POLAR ANGLE= 90.000 DEGREES
 DETECTOR AXIS AZIMUTH ANGLE= 180.000 DEGREES

GROUND PLANE PARAMETERS
 ISOTROPIC REFLECTIVITY GROUND PLANE
 GROUND PLANE ALBEDO, ALBG, = .500

CLOUD PARAMETERS
 ELLIPSOID PRINCIPAL XYZ HALF-AXES(KM) = .1000 .2000 .1000
 EULER ANGLES THE. PRE. OF ELLIPSOID(DEC) = .0000 .0000 .0000
 OPTICAL DEPTHS ALONG ELLIPSOID XYZ AXES = .8000 1.6000 .8000

STEADY STATE POWER TO DETECTOR, FOR UNIT SOURCE POWER

ORDER	STEADY STATE POWER	NUMBER OF PHOTONS
0	.27973-001	.50000+004
1	.66398-009	.50000+004
2	.90441-011	.49990+004
3	.18337-012	.50000+004
4	.12166-013	.50000+004
5	.58910-015	.500000+004
TOTAL	.27973-001	

POWER INTO DETECTOR FOR 1 PULSE(S) OF DIFFERENT LENGTH

PULSE NUMBER 1 HAS LENGTH .2667+000 MICROSECONDS
 DETECTOR RESPONSE CUTOFF TIME FOR PULSE NUMBER 1 IS .7067+001 MICROSECONDS

DETECTOR RESPONSE, POWER AS A FUNCTION OF TIME, FOR UNIT PULSE POWER

TIME	POWER FROM EACH ORDER				
	0	1	2	3	4
.0000	.0000	.0000	.0000	.0000	.0000
.1333+000	.2797-001	.6640-009	.9044-011	.1833-012	.0000
.2667+000	.2797-001	.6640-009	.9044-011	.1833-012	.0000
.4000+000	.0000	.0000	.1887-015	.5633-016	.2205-013
.5333+000	.0000	.0000	.2850-017	.4168-016	.2326-013
.6667+000	.0000	.0000	.0000	.1108-016	.1501-016
.8000+000	.3742-016	.0000	.0000	.0000	.0000

[illegible]

FINITE CLOUD RADIATIVE TRANSFER MODULE

-- RADIATIVE TRANSFER THROUGH FINITE CLOUD --

(XC,YC,ZC) = (.0000, .0000, .1000) KILOMETERS
 (XE,YE,ZE) = (.1000, .2000, .1000)
 (XB,YB,ZB) = (.1000, .2000, .0020)
 (XS,YS,ZS) = (-.2000, .0000, .0020)
 INDEXP = 5
 LAMBDA = 1.060 (MU)
 KAPPA = 4.000+000 (KM-1)
 OMEGA_0 = 1.000
 TAUBAR = 2.000+000
 THETA_0 = 80.0
 PHI_0 = 0 (DEGREES)
 RHO = .500
 LB_0 = 50.000 (W/M2-SR-MU)
 THPA = 9.8 (DEG.C)
 TMPC = 0
 LD = 0

**THERMAL CALCULATION OF PATH RADIANCE
 BB(LAMBDA,TPC) = 1.3071-013 W/M2-SR-MU
 BB(LAMBDA,THPA) = 1.3071-013 W/M2-SR-MU

PATH LENGTH (IN CLOUD)	TRANSMITTANCE	PATH RADIANCE (W/M2-SR-MU)	CONTRAST TRANSMITTANCE
.040	8.523-001	1.129-014	1.000+000

RUN NUMBER 3

**** EDSEL WARNING ****
VISIBILITY AND EXTINCTION = 0.0, VISIBILITY CHANGED TO 10.0 KM

INDIVIDUAL MODULES SELECTED		
	BEGINNING	ENDING
WAVENUMBER(CH**1)	1.167	1.167
WAVELENGTH(MICRONS)	8571.428	8571.428
FREQUENCY(GHZ)	35.000	35.000

DEFINITIONS OF METEOROLOGICAL CLASSES

470

ISRAELI CLIMATE FOR EUROPEAN LOWLANDS

VISIBILITY
9.30 KM

NEAR MILLIMETER WAVE MODULE

TEMPERATURE	5.300	DEGREES C
RELATIVE HUMIDITY	1012.900	HPa
AIR DENSITY	6.300	M**3
WIND SPEED	5.000	M/HR
SUN ELEVATION	.000	DG/HR
SLR FLUX	35.000	MJ/M ²
SLR PATH LENGTH	.403	KM
GSS ABSORPTION	.683E+001	DB/KM
FOG EXTINCTION	.735E+001	DB/KM
RAIN EXTINCTION	.735E+001	DB/KM
SNOW EXTINCTION	.0000	DB/KM
TRANSMISSION	.841E+000	
FOG BACKSCATTER	.772E+010	M**2/M***3
RAIN BACKSCATTER	.702E+010	M**2/M***3
SNOW BACKSCATTER	.0000	M**2/M***3
TOTAL BACKSCATTER	.102E+003	M**2/M***3

[illegible]

COMBINED TRANSMISSION FOR THE SELECTED MODULES = .5195+000

RUN NUMBER 4

**** EOSAEL WARNING ****
VISIBILITY AND EXTINCTION = 0.0, VISIBILITY CHANGED TO 10.0 KM

INDIVIDUAL MODULES SELECTED
SPOT
LT4M

	BEGINNING	ENDING
WAVENUMBER(CM**-1)	2010.000	2710.000
WAVELENGTH(MICRONS)	3.690	4.975
FREQUENCY(GHZ)	60300.000	81300.000

DEFINITIONS OF METEOROLOGICAL CLASSES

474

ISRAEL CLIMATOLOGY FOR EUROPEAN LOWLANDS									
CLASS NO.	FREQCY CLASS	MEAN TEMP (C)	MEAN DP (C)	MEAN RH (GM/CCU, H)	MEAN PRESS (MB)	MEAN/STDEV UNDEL (HRS)	MEAN/STDEV CLOUDR (%)	HT 03-09 (LST)	FREQCY CLASS
1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7
8	8	8	8	8	8	8	8	8	8
9	9	9	9	9	9	9	9	9	9
10	10	10	10	10	10	10	10	10	10
11	11	11	11	11	11	11	11	11	11
12	12	12	12	12	12	12	12	12	12
13	13	13	13	13	13	13	13	13	13
14	14	14	14	14	14	14	14	14	14
15	15	15	15	15	15	15	15	15	15
16	16	16	16	16	16	16	16	16	16
17	17	17	17	17	17	17	17	17	17
18	18	18	18	18	18	18	18	18	18
19	19	19	19	19	19	19	19	19	19
20	20	20	20	20	20	20	20	20	20
21	21	21	21	21	21	21	21	21	21
22	22	22	22	22	22	22	22	22	22
23	23	23	23	23	23	23	23	23	23
24	24	24	24	24	24	24	24	24	24
25	25	25	25	25	25	25	25	25	25
26	26	26	26	26	26	26	26	26	26
27	27	27	27	27	27	27	27	27	27
28	28	28	28	28	28	28	28	28	28
29	29	29	29	29	29	29	29	29	29
30	30	30	30	30	30	30	30	30	30
31	31	31	31	31	31	31	31	31	31
32	32	32	32	32	32	32	32	32	32
33	33	33	33	33	33	33	33	33	33
34	34	34	34	34	34	34	34	34	34
35	35	35	35	35	35	35	35	35	35
36	36	36	36	36	36	36	36	36	36
37	37	37	37	37	37	37	37	37	37
38	38	38	38	38	38	38	38	38	38
39	39	39	39	39	39	39	39	39	39
40	40	40	40	40	40	40	40	40	40
41	41	41	41	41	41	41	41	41	41
42	42	42	42	42	42	42	42	42	42
43	43	43	43	43	43	43	43	43	43
44	44	44	44	44	44	44	44	44	44
45	45	45	45	45	45	45	45	45	45
46	46	46	46	46	46	46	46	46	46
47	47	47	47	47	47	47	47	47	47
48	48	48	48	48	48	48	48	48	48
49	49	49	49	49	49	49	49	49	49
50	50	50	50	50	50	50	50	50	50
51	51	51	51	51	51	51	51	51	51
52	52	52	52	52	52	52	52	52	52
53	53	53	53	53	53	53	53	53	53
54	54	54	54	54	54	54	54	54	54
55	55	55	55	55	55	55	55	55	55
56	56	56	56	56	56	56	56	56	56
57	57	57	57	57	57	57	57	57	57
58	58	58	58	58	58	58	58	58	58
59	59	59	59	59	59	59	59	59	59
60	60	60	60	60	60	60	60	60	60
61	61	61	61	61	61	61	61	61	61
62	62	62	62	62	62	62	62	62	62
63	63	63	63	63	63	63	63	63	63
64	64	64	64	64	64	64	64	64	64
65	65	65	65	65	65	65	65	65	65
66	66	66	66	66	66	66	66	66	66
67	67	67	67	67	67	67	67	67	67
68	68	68	68	68	68	68	68	68	68
69	69	69	69	69	69	69	69	69	69
70	70	70	70	70	70	70	70	70	70
71	71	71	71	71	71	71	71	71	71
72	72	72	72	72	72	72	72	72	72
73	73	73	73	73	73	73	73	73	73
74	74	74	74	74	74	74	74	74	74
75	75	75	75	75	75	75	75	75	75
76	76	76	76	76	76	76	76	76	76
77	77	77	77	77	77	77	77	77	77
78	78	78	78	78	78	78	78	78	78
79	79	79	79	79	79	79	79	79	79
80	80	80	80	80	80	80	80	80	80
81	81	81	81	81	81	81	81	81	81
82	82	82	82	82	82	82	82	82	82
83	83	83	83	83	83	83	83	83	83
84	84	84	84	84	84	84	84	84	84
85	85	85	85	85	85	85	85	85	85
86	86	86	86	86	86	86	86	86	86
87	87	87	87	87	87	87	87	87	87
88	88	88	88	88	88	88	88	88	88
89	89	89	89	89	89	89	89	89	89
90	90	90	90	90	90	90	90	90	90
91	91	91	91	91	91	91	91	91	91
92	92	92	92	92	92	92	92	92	92
93	93	93	93	93	93	93	93	93	93
94	94	94	94	94	94	94	94	94	94
95	95	95	95	95	95	95	95	95	95
96	96	96	96	96	96	96	96	96	96
97	97	97	97	97	97	97	97	97	97
98	98	98	98	98	98	98	98	98	98
99	99	99	99	99	99	99	99	99	99
100	100	100	100	100	100	100	100	100	100

SPOT CONTRAST MODULE

SPOT CONTROL CARDS READ FOR THIS RUN:

ENVR 3000+001 2000+001 2000+001 4000+001 1000+001 1000+001
 EATS 1000+001 2830+003 9500+000 2950+003 0000 0000
 ATN 6500+002 0000 0000 0000 0000 0000
 TARG 4500+000 4500+002 9000+002 4500+002 0000 0000
 REFL 5000+000 5000+000 0000 5000+001 0000 0000
 SENS 2000+002 9000+002 2700+003 1000+001 0000 0000
 GO 0000 0000 0000 0000 0000 0000

DIVISION LIMITS CHANGED FROM 40.000 TO 60.000

INPUT RESPONSE FUNCTION

WAVELENGTH R FUNCTION

3500+001 7900+000
 3600+001 8000+000
 3700+001 8100+000
 3800+001 8200+000
 3900+001 8300+000
 4000+001 8400+000
 4100+001 8500+000
 4200+001 8600+000
 4300+001 8700+000
 4400+001 8800+000
 4500+001 8900+000
 4600+001 9000+000
 4700+001 9100+000
 4800+001 9200+000
 4900+001 9300+000
 5000+001 9400+000

SPOT DIAGNOSTIC MESSAGES FOLLOW:

1. NO DIRECT SUNLIGHT INCIDENT WITHIN RECEIVER'S FIELD OF VIEW.

SOURCE INTENSITIES			
WAVELENGTH (MICRONS)	WAVENUMBER (CM-1)	SUNLIGHT SOURCE STRENGTH (WATTS M-2 MICRON-1)	TARGET SOURCE STRENGTH (WATTS M-2 MICRON-1 SR-1)
9.751+000	2010	87.97+000	2.0519+000
4.370+000	2020	7.260+000	1.7239+000
4.333+000	2020	1.094+000	1.5271+000
4.309+000	2020	4.551+000	1.3055+000
4.284+000	2020	0.741+000	1.1387+000
4.260+000	2020	7.442+000	9.5228+001
4.236+000	2020	7.468+000	8.0289+001
4.212+000	2020	8.248+000	6.7321+001
4.188+000	2020	9.095+000	5.7621+001
4.164+000	2020	9.396+000	4.8223+001
4.140+000	2020	1.096+001	4.3940+001
4.116+000	2020		
4.092+000	2020		
4.068+000	2020		
4.044+000	2020		
4.020+000	2020		
3.996+000	2020		
3.972+000	2020		
3.948+000	2020		
3.924+000	2020		
3.900+000	2020		
3.876+000	2020		
3.852+000	2020		
3.828+000	2020		
3.804+000	2020		
3.780+000	2020		
3.756+000	2020		
3.732+000	2020		
3.708+000	2020		
3.684+000	2020		
3.660+000	2020		
3.636+000	2020		
3.612+000	2020		
3.588+000	2020		
3.564+000	2020		
3.540+000	2020		
3.516+000	2020		
3.492+000	2020		
3.468+000	2020		
3.444+000	2020		
3.420+000	2020		
3.396+000	2020		
3.372+000	2020		
3.348+000	2020		
3.324+000	2020		
3.300+000	2020		
3.276+000	2020		
3.252+000	2020		
3.228+000	2020		
3.204+000	2020		
3.180+000	2020		
3.156+000	2020		
3.132+000	2020		
3.108+000	2020		
3.084+000	2020		
3.060+000	2020		
3.036+000	2020		
3.012+000	2020		
2.988+000	2020		
2.964+000	2020		
2.940+000	2020		
2.916+000	2020		
2.892+000	2020		
2.868+000	2020		
2.844+000	2020		
2.820+000	2020		
2.796+000	2020		
2.772+000	2020		
2.748+000	2020		
2.724+000	2020		
2.700+000	2020		
2.676+000	2020		
2.652+000	2020		
2.628+000	2020		
2.604+000	2020		
2.580+000	2020		
2.556+000	2020		
2.532+000	2020		
2.508+000	2020		
2.484+000	2020		
2.460+000	2020		
2.436+000	2020		
2.412+000	2020		
2.388+000	2020		
2.364+000	2020		
2.340+000	2020		
2.316+000	2020		
2.292+000	2020		
2.268+000	2020		
2.244+000	2020		
2.220+000	2020		
2.196+000	2020		
2.172+000	2020		
2.148+000	2020		
2.124+000	2020		
2.100+000	2020		
2.076+000	2020		
2.052+000	2020		
2.028+000	2020		
2.004+000	2020		
1.980+000	2020		
1.956+000	2020		
1.932+000	2020		
1.908+000	2020		
1.884+000	2020		
1.860+000	2020		
1.836+000	2020		
1.812+000	2020		
1.788+000	2020		
1.764+000	2020		
1.740+000	2020		
1.716+000	2020		
1.692+000	2020		
1.668+000	2020		
1.644+000	2020		
1.620+000	2020		
1.596+000	2020		
1.572+000	2020		
1.548+000	2020		
1.524+000	2020		
1.500+000	2020		
1.476+000	2020		
1.452+000	2020		
1.428+000	2020		
1.404+000	2020		
1.380+000	2020		
1.356+000	2020		
1.332+000	2020		
1.308+000	2020		
1.284+000	2020		
1.260+000	2020		
1.236+000	2020		
1.212+000	2020		
1.188+000	2020		
1.164+000	2020		
1.140+000	2020		
1.116+000	2020		
1.092+000	2020		
1.068+000	2020		
1.044+000	2020		
1.020+000	2020		
0.996+000	2020		
0.972+000	2020		
0.948+000	2020		
0.924+000	2020		
0.900+000	2020		
0.876+000	2020		
0.852+000	2020		
0.828+000	2020		
0.804+000	2020		
0.780+000	2020		
0.756+000	2020		
0.732+000	2020		
0.708+000	2020		
0.684+000	2020		
0.660+000	2020		
0.636+000	2020		
0.612+000	2020		
0.588+000	2020		
0.564+000	2020		
0.540+000	2020		
0.516+000	2020		
0.492+000	2020		
0.468+000	2020		
0.444+000	2020		
0.420+000	2020		
0.396+000	2020		
0.372+000	2020		
0.348+000	2020		
0.324+000	2020		
0.300+000	2020		
0.276+000	2020		
0.252+000	2020		
0.228+000	2020		
0.204+000	2020		
0.180+000	2020		
0.156+000	2020		
0.132+000	2020		
0.108+000	2020		
0.084+000	2020		
0.060+000	2020		
0.036+000	2020		
0.012+000	2020		
0.000+000	2020		

COMPONENTS FOR RADIANCE FROM TARGET						
(WATTS M-2 MICRON-1 SR-1)						
WAVELENGTH (MICRONS)	WAVENUMBER (CM-1)	TARGET EMISSION	TARGET REFLECTANCE	PARTIAL ATMOSPHERIC EMISSION	PARTIAL PATH RADIANCE	TOTAL TARGET RADIANCE
4.9251+000	2010	8.367-001	3.550-003	5.55-001	2.645-004	1.532+000
4.809+000	2020	7.084-001	3.420-003	5.55-001	1.042-004	1.322+000
4.662+000	2040	6.180-001	3.180-003	5.55-001	4.879-004	1.111+000
4.444+000	2230	6.232-001	3.180-003	5.55-001	4.879-004	1.623-001
4.213+000	2370	6.510-004	3.180-003	5.55-001	4.879-004	1.623-001
4.112+000	2430	6.510-004	3.180-003	5.55-001	4.879-004	1.623-001
4.021+000	2480	6.510-004	3.180-003	5.55-001	4.879-004	1.623-001
3.925+000	2520	6.510-004	3.180-003	5.55-001	4.879-004	1.623-001
3.825+000	2560	6.510-004	3.180-003	5.55-001	4.879-004	1.623-001
3.725+000	2600	6.510-004	3.180-003	5.55-001	4.879-004	1.623-001
3.625+000	2670	6.510-004	3.180-003	5.55-001	4.879-004	1.623-001

COMPONENTS FOR BACKGROUND RADIANCE

(WATTS M-2 MICRON-1 SR-1)

WAVELENGTH (MICRONS)	WAVENUMBER (CM-1)	GROUND EMISSION	GROUND REFLECTANCE	TOTAL ATMOSPHERIC EMISSION	TOTAL PATH RADIANCE	TOTAL BACKGROUND RADIANCE
4.51+000	2010	.0000	.0000	1.433+000	1.350-004	1.647+000
4.59+000	2070	.0000	.0000	1.403+000	1.355-004	1.409+000
4.66+000	2130	.0000	.0000	1.203+000	1.203-003	1.207+000
4.74+000	2190	.0000	.0000	1.021+000	1.020-003	1.027+000
4.82+000	2250	.0000	.0000	8.321-001	1.025-003	9.721-001
4.90+000	2310	.0000	.0000	7.321-001	1.025-003	8.721-001
4.98+000	2370	.0000	.0000	6.161-001	1.025-003	7.611-001
5.06+000	2430	.0000	.0000	5.161-001	1.025-003	6.516-001
5.14+000	2490	.0000	.0000	4.321-001	1.025-003	5.723-001
5.22+000	2550	.0000	.0000	3.635-001	1.025-003	5.235-001
5.30+000	2610	.0000	.0000	3.249-001	1.025-003	4.655-001
5.38+000	2670	.0000	.0000	2.489-001	1.025-003	3.950-001

DIRECT SUNLIGHT			
(WATTS M-2 MICRON-1)			
WAVELENGTH (MICRONS)	WAVENUMBER (CM-1)	SUNLIGHT SOURCE STRENGTH	SUNLIGHT FLUX
4.751+000	2010	7.927+000	.0000
4.752+000	2000	7.927+000	.0000
4.753+000	2000	7.927+000	.0000
4.754+000	2000	7.927+000	.0000
4.755+000	2000	7.927+000	.0000
4.756+000	2000	7.927+000	.0000
4.757+000	2000	7.927+000	.0000
4.758+000	2000	7.927+000	.0000
4.759+000	2000	7.927+000	.0000
4.760+000	2000	7.927+000	.0000
4.761+000	2000	7.927+000	.0000
4.762+000	2000	7.927+000	.0000
4.763+000	2000	7.927+000	.0000
4.764+000	2000	7.927+000	.0000
4.765+000	2000	7.927+000	.0000
4.766+000	2000	7.927+000	.0000
4.767+000	2000	7.927+000	.0000
4.768+000	2000	7.927+000	.0000
4.769+000	2000	7.927+000	.0000
4.770+000	2000	7.927+000	.0000
4.771+000	2000	7.927+000	.0000
4.772+000	2000	7.927+000	.0000
4.773+000	2000	7.927+000	.0000
4.774+000	2000	7.927+000	.0000
4.775+000	2000	7.927+000	.0000
4.776+000	2000	7.927+000	.0000
4.777+000	2000	7.927+000	.0000
4.778+000	2000	7.927+000	.0000
4.779+000	2000	7.927+000	.0000
4.780+000	2000	7.927+000	.0000
4.781+000	2000	7.927+000	.0000
4.782+000	2000	7.927+000	.0000
4.783+000	2000	7.927+000	.0000
4.784+000	2000	7.927+000	.0000
4.785+000	2000	7.927+000	.0000
4.786+000	2000	7.927+000	.0000
4.787+000	2000	7.927+000	.0000
4.788+000	2000	7.927+000	.0000
4.789+000	2000	7.927+000	.0000
4.790+000	2000	7.927+000	.0000
4.791+000	2000	7.927+000	.0000
4.792+000	2000	7.927+000	.0000
4.793+000	2000	7.927+000	.0000
4.794+000	2000	7.927+000	.0000
4.795+000	2000	7.927+000	.0000
4.796+000	2000	7.927+000	.0000
4.797+000	2000	7.927+000	.0000
4.798+000	2000	7.927+000	.0000
4.799+000	2000	7.927+000	.0000
4.800+000	2000	7.927+000	.0000

AD-A114 417

ARMY ELECTRONICS RESEARCH AND DEVELOPMENT COMMAND WS--ETC F/G 4/1
PROGRAM LISTINGS FOR EOSAEL 80-B AND ANCILLARY CODES AGAUS AND --ETC(11)
FEB 82 R G STEINHOFF
ERADCOM/ASL-TR-0107-V2-SU

UNCLASSIFIED

NL

6 x 6

6 x 6

END
DATE
FILMED
6 82
DTIC

TOTAL RADIANCE

(WATTS M-2 MICRON-1 SR-1)

WAVELENGTH (MICRONS)	WAVENUMBER (CM-1)	TARGET	BACKGROUND	CONTRAST RATIO
4.9751+000	2010	1.5362+000	1.6447+000	-6.5959-002
4.8309+000	2070	1.3232+000	1.4094+000	-6.1179-002
4.6948+000	2130	1.1211+000	1.2059+000	-7.0270-002
4.5662+000	2190	9.6050-001	1.0274+000	-6.5111-002
4.4444+000	2250	8.4575-001	8.6716-001	-2.4691-002
4.3290+000	2310	7.3211-001	7.3217-001	-7.7684-005
4.2194+000	2370	6.1542-001	6.1611-001	-1.1165-003
4.1152+000	2430	5.1886-001	5.3016-001	-2.1316-002
4.0161+000	2490	4.6102-001	4.6672-001	9.4083-003
3.9216+000	2550	4.1435-001	4.7285-001	4.9959-002
3.8314+000	2610	3.7419-001	4.4785-001	7.9745-002
3.7453+000	2670	3.2256-001	3.6956-001	1.1420-001

DETECTOR-RESPONSE WAVELENGTH-INTEGRATED

(WATTS M-2 SR-1)

TARGET EMISSION	4.9117-002
TARGET REFLECTANCE	1.8157-003
PARTIAL ATMOSPHERIC EMISSION	4.3029-002
PARTIAL PATH RADIANCE	7.7395-005
TOTAL TARGET RADIANCE	9.4039-002
GROUND EMISSION	.0000
GROUND REFLECTANCE	.0000
TOTAL ATMOSPHERIC EMISSION	9.7352-002
TOTAL PATH RADIANCE	9.1328-004
TOTAL BACKGROUND RADIANCE	9.8265-002

CONTRAST	-4.3004-002
----------	-------------

DIRECT SUNLIGHT	.0000
(WATTS M-2)	

LT4M ATMOSPHERIC TRANSMISSION MODULE

PROGRAM WILL BE EXECUTED IN THE TRANSMISSION MODE

HORIZONTAL PATH, ALTITUDE = .002 KM, RANGE = .400 KM

MODEL ATMOSPHERE 4 = SUB-ARCTIC SUMMER

FREQUENCY RANGE $\nu_1 = 2010.0 \text{ CM}^{-1}$ TO $\nu_2 = 2710.0 \text{ CM}^{-1}$ FOR $\Delta\nu = 60.0 \text{ CM}^{-1}$ (3.69 - 4.98 MICRONS)

EQUIVALENT SEA LEVEL ABSORBER AMOUNTS

	WATER VAPOUR GM CM-2	CO2 ETC. KM	OZONE ATM CM	NITROGEN (CONT) KM	H2O (CONT) GM CM-2	MOL KM	SCAT	OZONE(U-V) ATM CM
$W(1-6,8) =$ $W(10) =$.355+000	.372+000	.905-003	.295+000	.595-002 .546-001	.380+000		.915-003
	NITRIC ACID	SO2	NH3	NO2				
$W(11-16) =$.000	.834-002	.779-002	.337-002	.433-002	.707-004		

FREQ	WAVELENGTH	H2O	CO2+	OZONE	N2 C	H2O C	HOL S	NITRIC	SO2	HN03	NO2	INTEGRATED	TOTAL	AEROSOL
CM-1	MICRONS	TRANS	TRANS	TRANS	TRANS	TRANS	TRANS	TRANS	TRANS	TRANS	TRANS	ABSORPTION	TRANS	TRANS
2010	4.9751	.6090	.9987	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	12.6982	.5767	.94820
2070	4.8309	.7608	.9989	1.0000	1.0000	.8220	1.0000	1.0000	1.0000	1.0000	1.0000	38.8084	.5648	.94820
2130	4.6949	.8649	.9996	1.0000	1.0000	.9066	1.0000	1.0000	1.0000	1.0000	1.0000	54.9334	.7313	.94820
2190	4.5662	.9398	.9999	1.0000	.9990	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	70.8646	.7345	.94820
2250	4.4444	.9367	.9999	1.0000	.9915	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	113.9851	.2813	.94820
2310	4.3290	.9769	.9999	1.0000	.9807	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	173.9270	.0010	.94820
2370	4.2194	.9925	.9999	1.0000	.9652	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	233.0067	.0153	.94820
2430	4.1152	.9977	.9999	1.0000	.9723	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	238.5729	.9006	.94820
2490	4.0161	.9986	.9999	1.0000	.9890	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	243.8880	.3181	.94820
2550	3.9216	.9903	.9976	1.0000	.9974	1.0000	1.0000	1.0000	.9971	1.0000	1.0000	249.4692	.9070	.94820
2610	3.8314	.9963	.9985	1.0000	.9993	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	255.0312	.9073	.94820
2670	3.7453	.9383	.9981	1.0000	.9996	.9916	1.0000	1.0000	1.0000	1.0000	1.0000	262.2208	.8802	.94820

WAVELENGTH AND SENSOR INTEGRATED TRANSMISSION = .4711-001

INTEGRATED ASORPTION FROM 2010 TO 2730 CM-1 = 262.22, AVERAGE TRANSMITTANCE = .6358

LT4M ATMOSPHERIC TRANSMISSION MODULE
PROGRAM WILL BE EXECUTED IN THE EMISSION MODE

HORIZONTAL PATH, ALTITUDE = .002 KM, RANGE = .400 KM

MODEL ATMOSPHERE 4 = SUB-ARCTIC SUMMER

FREQUENCY RANGE V1= 2010.0 CM-1 TO V2= 2710.0 CM-1 FOR DV = 60.0 CM-1 < 3.69 - 4.98 MICRONS >

EQUIVALENT SEA LEVEL ABSORBER AMOUNTS

	WATER VAPOUR GM CM-2	CO2 ETC. KM	OZONE ATH CM	NITROGEN (CONT) KM	H2O (CONT) GM CM-2	MOL SCAT KM	OZONE(U-V) ATH CM
W(1-6.8)=	.355+000	.372+000	.905-003	.295+000	.595-002	.380+000	.915-003
W(10)=					.546-001		

	NITRIC ACID	SO2	NH3	NO2		
W(11-16)=	.000	.834-002	.779-002	.337-002	.433-002	.707-004

CUMULATIVE ABSORBER AMOUNTS FOR THE ATMOSPHERIC PATH

	H2O	CO2+	O3	N2	H2O C	MOL S	O3 UV	H2O C	HN03	SO2	NH3	NO2	TAVE
1	.355+000	.372+000	.905-003	.295+000	.595-002	.380+000	.915-003	.546-001	.000	.834-002	.779-002	.337-002	287.000

FR(CM-1)	WVL(MICRON)	RADIANCE(WATTS/CM2-STER-XXX)		INTEGRAL	TRANS	AERO TRAN	
		PER CM-1	PER MICRON			EXTN	TRAN
2010.0	4.975124	.17216-006	.69555-004	.51648-005	.576727	.94820	.98942
2070.0	4.830918	.14311-006	.61320-004	.13751-004	.564830	.94820	.98942
2130.0	4.694836	.71276-007	.32337-004	.18028-004	.731250	.94820	.98942
2190.0	4.566210	.56657-007	.27173-004	.21427-004	.734479	.94820	.98942
2250.0	4.444444	.12310-006	.62320-004	.28813-004	.281320	.94820	.98942
2310.0	4.329004	.13708-006	.73146-004	.37038-004	.000967	.94820	.98942
2370.0	4.219409	.10801-006	.60666-004	.43518-004	.015339	.94820	.98942
2430.0	4.115226	.87026-008	.51388-005	.44040-004	.900563	.94820	.98942
2490.0	4.016064	.57098-008	.35402-005	.44383-004	.918683	.94820	.98942
2550.0	3.921569	.51549-008	.33520-005	.44692-004	.906979	.94820	.98942
2610.0	3.831418	.40774-008	.27776-005	.44937-004	.907301	.94820	.98942
2670.0	3.745318	.41768-008	.29776-005	.45188-004	.880173	.94820	.98942

WAVELENGTH AND SENSOR INTEGRATED TRANSMISSION = .4711-001

RADMIN 2610.000 .40774-008
 RADMAX 2010.000 .17216-006

INTEGRATED ASORPTION FROM 2010 TO 2730 CM-1 = 262.22, AVERAGE TRANSMITTANCE = .6358
 INTEGRATED RADIANCE = .45188-004 WATT CM -2 SR

COMBINED TRANSMISSION FOR THE SELECTED MODULES = .4711-001

END EOSAEL RUN

ELECTRO-OPTICS DISTRIBUTION LIST

Commander
US Army Aviation School
Fort Rucker, AL 36362

Commander
US Army Aviation Center
ATTN: ATZQ-D-MA (Mr. Oliver N. Heath)
Fort Rucker, AL 36362

Commander
US Army Aviation Center
ATTN: ATZQ-D-MS (Mr. Donald Wagner)
Fort Rucker, AL 36362

NASA/Marshall Space Flight Center
ATTN: ES-83 (Otha H. Vaughan, Jr.)
Huntsville, AL 35812

NASA/Marshall Space Flight Center
Atmospheric Sciences Division
ATTN: Code ES-81 (Dr. William W. Vaughan)
Huntsville, AL 35812

Nichols Research Corporation
ATTN: Dr. Lary W. Pinkley
4040 South Memorial Parkway
Huntsville, AL 35802

John M. Hobbie
c/o Kentron International
2003 Byrd Spring Road
Huntsville, AL 35802

Mr. Ray Baker
Lockheed-Missile & Space Company
4800 Bradford Blvd
Huntsville, AL 35807

Commander
US Army Missile Command
ATTN: DRSMI-OG (Mr. Donald R. Peterson)
Redstone Arsenal, AL 35809

Commander
US Army Missile Command
ATTN: DRSMI-OGA (Dr. Bruce W. Fowler)
Redstone Arsenal, AL 35809

Commander
US Army Missile Command
ATTN: DRSMI-REL (Dr. George Emmons)
Redstone Arsenal, AL 35809

Commander
US Army Missile Command
ATTN: DRSMI-REO (Huey F. Anderson)
Redstone Arsenal, AL 35809

Commander
US Army Missile Command
ATTN: DRSMI-REO (Mr. Maxwell W. Harper)
Redstone Arsenal, AL 35809

Commander
US Army Missile Command
ATTN: DRSMI-REO (Mr. Gene Widenhofer)
Redstone Arsenal, AL 35809

Commander
US Army Missile Command
ATTN: DRSMI-RHC (Dr. Julius Q. Lilly)
Redstone Arsenal, AL 35809

Commander
US Army Missile Command
Redstone Scientific Information Center
ATTN: DRSMI-RPRD (Documents Section)
Redstone Arsenal, AL 35809

Commander
US Army Missile Command
ATTN: DRSMI-RRA (Dr. Oskar Essenwanger)
Redstone Arsenal, AL 35809

Commander
US Army Missile Command
ATTN: DRSMI-RRO (Mr. Charles Christensen)
Redstone Arsenal, AL 35809

Commander
US Army Missile Command
ATTN: DRSMI-RRO (Dr. George A. Tanton)
Redstone Arsenal, AL 35809

Commander
US Army Communications Command
ATTN: CC-OPS-PP
Fort Huachuca, AZ 85613

Commander
US Army Intelligence Center & School
ATTN: ATSI-CD-CS (Mr. Richard G. Cundy)
Fort Huachuca, AZ 85613

Commander
US Army Intelligence Center & School
ATTN: ATSI-CD-MD (Mr. Harry Wilder)
Fort Huachuca, AZ 85613

Commander
US Army Intelligence Center & School
ATTN: ATSI-CS-C (2LT Coffman)
Fort Huachuca, AZ 85613

Commander
US Army Yuma Proving Ground
ATTN: STEYP-MSA-TL
Bldg 2105
Yuma, AZ 85364

Northrop Corporation
Electro-Mechanical Division
ATTN: Dr. Richard D. Tooley
500 East Orangethorpe Avenue
Anaheim, CA 92801

Commander
Naval Weapons Center
ATTN: Code 3918 (Dr. Alexis Shlanta)
China Lake, CA 93555

Hughes Helicopters
Army Advanced Attack Helicopter Weapons
ATTN: Mr. Charles R. Hill
Centinela and Teale Streets
Bldg 305, MS T-73A
Culter City, CA 90230

Commander
US Army Combat Developments
Experimentation Command
ATTN: ATEC-PL-M (Mr. Gary G. Love)
Fort Ord, CA 93941

SRI International
ATTN: K2060/Dr. Edward E. Uthe
333 Ravenswood Avenue
Menlo Park, CA 94025

SRI International
ATTN: Mr. J. E. Van der Laan
333 Ravenswood Avenue
Menlo Park, CA 94025

Joane May
Naval Environmental Prediction
Research Facility (NEPRF)
ATTN: Library
Monterey, CA 93940

Sylvania Systems Group,
Western Division
GTE Products Corporation
ATTN: Technical Reports Library
P.O. Box 205
Mountain View, CA 94042

Sylvania Systems Group
Western Division
GTE Products Corporation
ATTN: Mr. Lee W. Carrier
P.O. Box 188
Mountain View, CA 94042

Pacific Missile Test Center
Geophysics Division
ATTN: Code 3250-3 (R. de Violini)
Point Mugu, CA 93042

Pacific Missile Test Center
Geophysics Division
ATTN: Code 3253 (Terry E. Battalino)
Point Mugu, CA 93042

Effects Technology Inc.
ATTN: Mr. John D. Carlyle
5383 Hollister Avenue
Santa Barbara, CA 93111

Commander
Naval Ocean Systems Center
ATTN: Code 532 (Dr. Juergen Richter)
San Diego, CA 92152

Commander
Naval Ocean Systems Center
ATTN: Code 5322 (Mr. Herbert G. Hughes)
San Diego, CA 92152

Commander
Naval Ocean Systems Center
ATTN: Code 4473 (Tech Library)
San Diego, CA 92152

The RAND Corporation
ATTN: Ralph Huschke
1700 Main Street
Santa Monica, CA 90406

Particle Measuring Systems, Inc.
ATTN: Dr. Robert G. Knollenberg
1855 South 57th Court
Boulder, CO 80301

US Department of Commerce
National Oceanic and Atmospheric Admin
Environmental Research Laboratories
ATTN: Library, R-51, Technical Reports
325 Broadway
Boulder, CO 80303

US Department of Commerce
National Oceanic and Atmospheric Admin
Environmental Research Laboratories
ATTN: R45X3 (Dr. Vernon E. Derr)
Boulder, CO 80303

US Department of Commerce
National Telecommunications and
Information Administration
Institute for Telecommunication Sciences
ATTN: Code 1-3426 (Dr. Hans J. Liebe)
Boulder, CO 80303

AFATL/DLODL
Technical Library
Eglin AFB, FL 32542

Commanding Officer
Naval Training Equipment Center
ATTN: Technical Information Center
Orlando, FL 32813

Georgia Institute of Technology
Engineering Experiment Station
ATTN: Dr. Robert W. McMillan
Atlanta, GA 30332

Georgia Institute of Technology
Engineering Experiment Station
ATTN: Dr. James C. Wiltse
Atlanta, GA 30332

Commandant
US Army Infantry Center
ATTN: ATSH-CD-MS-E (Mr. Robert McKenna)
Fort Benning, GA 31805

Commander
US Army Signal Center & Fort Gordon
ATTN: ATZHCD-CS
Fort Gordon, GA 30905

Commander
US Army Signal Center & Fort Gordon
ATTN: ATZHCD-O
Fort Gordon, GA 30905

USAFETAC/DNE
ATTN: Mr. Charles Glauber
Scott AFB, IL 62225

Commander
Air Weather Service
ATTN: AWS/DNDP (LTC Kit G. Cottrell)
Scott AFB, IL 62225

Commander
Air Weather Service
ATTN: AWS/DOOF (MAJ Robert Wright)
Scott AFB, IL 62225

Commander
US Army Combined Arms Center
& Ft. Leavenworth
ATTN: ATZLCA-CAA-Q (Mr. H. Kent Pickett)
Fort Leavenworth, KS 66027

Commander
US Army Combined Arms Center
& Ft. Leavenworth
ATTN: ATZLCA-SAN (Robert DeKinder, Jr.)
Fort Leavenworth, KS 66027

Commander
US Army Combined Arms Center
& Ft. Leavenworth
ATTN: ATZLCA-SAN (Mr. Kent I. Johnson)
Fort Leavenworth, KS 66027

Commander
US Army Combined Arms Center
& Ft. Leavenworth
ATTN: ATZLCA-WE (LTC Darrell Holland)
Fort Leavenworth, KS 66027

President
USAARENBD
ATTN: ATZK-AE-TA (Dr. Charles R. Leake)
Fort Knox, KY 40121

Commander
US Army Armor Center and Fort Knox
ATTN: ATZK-CD-MS
Fort Knox, KY 40121

Commander
US Army Armor Center and Fort Knox
ATTN: ATZK-CD-SD
Fort Knox, KY 40121

Aerodyne Research Inc.
ATTN: Dr. John F. Ebersole
Crosby Drive
Bedford, MA 01730

Commander
Air Force Geophysics Laboratory
ATTN: OPA (Dr. Robert W. Fenn)
Hanscom AFB, MA 01731

Commander
Air Force Geophysics Laboratory
ATTN: OPI (Dr. Robert A. McClatchey)
Hanscom AFB, MA 01731

Massachusetts Institute of Technology
Lincoln Laboratory
ATTN: Dr. T. J. Goblick, B-370
P.O. Box 73
Lexington, MA 02173

Massachusetts Institute of Technology
Lincoln Laboratory
ATTN: Dr. Michael Gruber
P.O. Box 73
Lexington, MA 02173

Raytheon Company
Equipment Division
ATTN: Dr. Charles M. Sonnenschein
430 Boston Post Road
Wayland, MA 01778

Commander
US Army Ballistic Research Laboratory/
ARRADCOM
ATTN: DRDAR-BLB (Mr. Richard McGee)
Aberdeen Proving Ground, MD 21005

Commander/Director
Chemical Systems Laboratory
US Army Armament Research
& Development Command
ATTN: DRDAR-CLB-PS (Dr. Edward Stuebing)
Aberdeen Proving Ground, MD 21010

Commander/Director
Chemical Systems Laboratory
US Army Armament Research
& Development Command
ATTN: DRDAR-CLB-PS (Mr. Joseph Vervier)
Aberdeen Proving Ground, MD 21010

Commander/Director
Chemical Systems Laboratory
US Army Armament Research
& Development Command
ATTN: DRDAR-CLY-A (Mr. Ronald Pennsyle)
Aberdeen Proving Ground, MD 21010

Commander
US Army Ballistic Research Laboratory/
ARRADCOM
ATTN: DRDAR-TSB-S (STINFO)
Aberdeen Proving Ground, MD 21005

Commander
US Army Electronics Research
& Development Command
ATTN: DRDEL-CCM (W. H. Pepper)
Adelphi, MD 20783

Commander
US Army Electronics Research
& Development Command
ATTN: DRDEL-CG/DRDEL-DC/DRDEL-CS
2800 Powder Mill Road
Adelphi, MD 20783

Commander
US Army Electronics Research
& Development Command
ATTN: DRDEL-CT
2800 Powder Mill Road
Adelphi, MD 20783

Commander
US Army Electronics Research
& Development Command
ATTN: DRDEL-PAO (M. Singleton)
2800 Powder Mill Road
Adelphi, MD 20783

Project Manager
Smoke/Obscurants
ATTN: DRDPM-SMK
(Dr. Anthony Van de Wal, Jr.)
Aberdeen Proving Ground, MD 21005

Project Manager
Smoke/Obscurants
ATTN: DRDPM-SMK-T (Mr. Sidney Gerard)
Aberdeen Proving Ground, MD 21005

Commander
US Army Test & Evaluation Command
ATTN: DRSTE-AD-M (Mr. Warren M. Baity)
Aberdeen Proving Ground, MD 21005

Commander
US Army Test & Evaluation Command
ATTN: DRSTE-AD-M (Dr. Norman E. Pentz)
Aberdeen Proving Ground, MD 21005

Director
US Army Materiel Systems Analysis Activity
ATTN: DRXSY-AAM (Mr. William Smith)
Aberdeen Proving Ground, MD 21005

Director
US Army Materiel Systems Analysis Activity
ATTN: DRXSY-CS (Mr. Philip H. Beavers)
Aberdeen Proving Ground, MD 21005

Director
US Army Materiel Systems Analysis Activity
ATTN: DRXSY-GB (Wilbur L. Warfield)
Aberdeen Proving Ground, MD 21005

Director
US Army Materiel Systems Analysis Activity
ATTN: DRXSY-GP (Mr. Fred Campbell)
Aberdeen Proving Ground, MD 21005

Director
US Army Materiel Systems Analysis Activity
ATTN: DRXSY-GP (H. Stamper)
Aberdeen Proving Grounds, MD 21005

Director
US Army Materiel Systems Analysis Activity
ATTN: DRXSY-GS
(Mr. Michael Starks/Mr. Julian Chernick)
Aberdeen Proving Ground, MD 21005

Director
US Army Materiel Systems Analysis Activity
ATTN: DRXSY-J (Mr James F. O'Bryon)
Aberdeen Proving Ground, MD 21005

Director
US Army Materiel Systems Analysis Activity
ATTN: DRXSY-LM (Mr. Robert M. Marchetti)
Aberdeen Proving Ground, MD 21005

Commander
Harry Diamond Laboratories
ATTN: Dr. William W. Carter
2800 Powder Mill Road
Adelphi, MD 20783

Commander
Harry Diamond Laboratories
ATTN: DELHD-R-CM (Mr. Robert McCoskey)
2800 Powder Mill Road
Adelphi, MD 20783

Commander
Harry Diamond Laboratories
ATTN: DELHD-R-CM-NM (Dr. Robert Humphrey)
2800 Powder Mill Road
Adelphi, MD 20783

Commander
Harry Diamond Laboratories
ATTN: DELHD-R-CM-NM (Dr. Z. G. Sztankay)
2800 Powder Mill Road
Adelphi, MD 20783

Commander
Harry Diamond Laboratories
ATTN: DELHD-R-CM-NM (Dr. Joseph Nemerich)
2800 Powder Mill Road
Adelphi, MD 20783

Commander
Air Force Systems Command
ATTN: WER (Mr. Richard F. Picanso)
Andrews AFB, MD 20334

Martin Marietta Laboratories
ATTN: Jar Mo Chen
1450 South Rolling Road
Baltimore, MD 21227

Commander
US Army Concepts Analysis Agency
ATTN: CSCA-SMC (Mr. Hal E. Hock)
8120 Woodmont Avenue
Bethesda, MD 20014

Director
National Security Agency
ATTN: R52/Dr. Douglas Woods
Fort George G. Meade, MD 20755

Chief
Intelligence Materiel Development
& Support Office
US Army Electronic Warfare Laboratory
ATTN: DELEW-I (LTC Kenneth E. Thomas)
Fort George G. Meade, MD 20755

The Johns Hopkins University
Applied Physics Laboratory
ATTN: Dr. Michael J. Lun
John Hopkins Road
Laurell, MD 20810

Dr. Stephen T. Hanley
1720 Rhodesia Avenue
Oxon Hill, MD 20022

Science Applications Inc.
ATTN: Mr. G. D. Currie
15 Research Drive
Ann Arbor, MI 48103

Science Applications Inc.
ATTN: Dr. Robert E. Turner
15 Research Drive
Ann Arbor, MI 48103

Commander
US Army Tank-Automotive Research
& Development Command
ATTN: DRDTA-ZSC (Mr. Harry Young)
Warren, MI 48090

Commander
US Army Tank Automotive Research
& Development Command
ATTN: DRDTA-ZSC (Mr. Wallace Mick, Jr.)
Warren, MI 48090

Dr. A. D. Belmont
Research Division
Control Data Corporation
P.O. Box 1249
Minneapolis, MN 55440

Director
US Army Engr Waterways Experiment Station
ATTN: WESEN (Mr. James Mason)
P.O. Box 631
Vicksburg, MS 39180

Dr. Jerry Davis
Department of Marine, Earth
and Atmospheric Sciences
North Carolina State University
Raleigh, NC 27650

Commander
US Army Research Office
ATTN: DRXRO-GS (Dr. Leo Alpert)
P.O. Box 12211
Research Triangle Park, NC 27709

Commander
US Army Research Office
ATTN: DRXRO-PP (Brenda Mann)
P.O. Box 12211
Research Triangle Park, NC 27709

Commander
US Army Cold Regions Research
& Engineering Laboratory
ATTN: CRREL-RD (Dr. K. F. Sterrett)
Hanover, NH 03755

Commander/Director
US Army Cold Regions Research
& Engineering Laboratory
ATTN: CRREL-RG (Mr. George Aitken)
Hanover, NH 03755

Commander
US Army Cold Regions Research
& Engineering Laboratory
ATTN: CRREL-RG (Mr. Roger H. Berger)
Hanover, NH 03755

Commander
US Army Armament Research
& Development Command
ATTN: DRDAR-AC (Mr. James Greenfield)
Dover, NJ 07801

Commander
US Army Armament Research
& Development Command
ATTN: DRDAR-TSS (Bldg #59)
Dover, NJ 07801

Commander
US Army Armament Research
& Development Command
ATTN: DRCPM-CAWS-EI (Mr. Peteris Jansons)
Dover, NJ 07801

Commander
US Army Armament Research
& Development Command
ATTN: DRCPM-CAWS-EI (Mr. G. H. Waldron)
Dover, NJ 07801

Deputy Joint Project Manager
for Navy/USMC SAL GP
ATTN: DRCPM-CAWS-NV (CPT Joseph Miceli)
Dover, NJ 07801

Commander/Director
US Army Combat Surveillance & Target
Acquisition Laboratory
ATTN: DELCS-I (Mr. David Longinotti)
Fort Monmouth, NJ 07703

Commander/Director
US Army Combat Surveillance & Target
Acquisition Laboratory
ATTN: DELCS-PE (Mr. Ben A. Di Campi)
Fort Monmouth, NJ 07703

Commander/Director
US Army Combat Surveillance & Target
Acquisition Laboratory
ATTN: DELCS-R-S (Mr. Donald L. Fofani)
Fort Monmouth, NJ 07703

Director
US Army Electronics Technology &
Devices Laboratory
ATTN: DELET-DD (S. Danko)
Fort Monmouth, NJ 07703

Project Manager
FIREFINDER/REMBASS
ATTN: DRCPM-FFR-TM (Mr. John M. Bialo)
Fort Monmouth, NJ 07703

Commander
US Army Electronics Research
& Development Command
ATTN: DRDEL-SA (Dr. Walter S. McAfee)
Fort Monmouth, NJ 07703

OLA, 2WS (MAC)
Holloman AFB, NM 88330
Commander
Air Force Weapons Laboratory
ATTN: AFWL/WE (MAJ John R. Elrick)
Kirtland, AFB, NM 87117

Director
USA TRADOC Systems Analysis Activity
ATTN: ATAA-SL
White Sands Missile Range, NM 88002

Director
USA TRADOC Systems Analysis Activity
ATTN: ATAA-SL (Dolores Anguiano)
White Sands Missile Range, NM 88002

Director
USA TRADOC Systems Analysis Activity
ATTN: ATAA-TDB (Mr. Louie Dominguez)
White Sands Missile Range, NM 88002

Director
USA TRADOC Systems Analysis Activity
ATTN: ATAA-TDB (Mr. William J. Leach)
White Sands Missile Range, NM 88002

Director
USA TRADOC Systems Analysis Activity
ATTN: ATAA-TGP (Mr. Roger F. Willis)
White Sands Missile Range, NM 88002

Director
Office of Missile Electronic Warfare
ATTN: DELEW-M-STO (Dr. Steven Kovel)
White Sands Missile Range, NM 88002

Office of the Test Director
Joint Services EO GW CM Test Program
ATTN: DRXDE-TD (Mr. Weldon Findley)
White Sands Missile Range, NM 88002

Commander
US Army White Sands Missile Range
ATTN: STEWS-PT-AL (Laurel B. Saunders)
White Sands Missile Range, NM 88002

Commander
US Army R&D Coordinator
US Embassy - Bonn
Box 165
APO New York 09080

Grumman Aerospace Corporation
Research Department - MS A08-35
ATTN: John E. A. Selby
Bethpage, NY 11714

Rome Air Development Center
ATTN: Documents Library
TSLD (Bette Smith)
Griffiss AFB, NY 13441

Dr. Roberto Vaglio-Laurin
Faculty of Arts and Science
Dept. of Applied Science
26-36 Stuyvesant Street
New York, NY 10003

Air Force Wright Aeronautical Laboratories/
Avionics Laboratory
ATTN: AFWAL/AARI-3 (Mr. Harold Geltmacher)
Wright-Patterson AFB, OH 45433

Air Force Wright Aeronautical Laboratories/
Avionics Laboratory
ATTN: AFWAL/AARI-3 (CPT William C. Smith)
Wright-Patterson AFB, OH 45433

Commandant
US Army Field Artillery School
ATTN: ATSF-CF-R (CPT James M. Watson)
Fort Sill, OK 73503

Commandant
US Army Field Artillery School
ATTN: ATSF-CD-MS
Fort Sill, OK 73503

Commandant
US Army Field Artillery School
ATTN: ATSF-CF-R
Fort Sill, OK 73503

Commandant
US Army Field Artillery School
ATTN: NOAA Liaison Officer
(CDR Jeffrey G. Carlen)
Fort Sill, OK 73503

Commandant
US Army Field Artillery School
Morris Swett Library
ATTN: Reference Librarian
Fort Sill, OK 73503

Commander
Naval Air Development Center
ATTN: Code 301 (Mr. George F. Eck)
Warminster, PA 18974

The University of Texas at El Paso
Electrical Engineering Department
ATTN: Dr. Joseph H. Pierluissi
El Paso, TX 79968

Commandant
US Army Air Defense School
ATTN: ATSA-CD-SC-A (CPT Charles T. Thorn)
Fort Bliss, TX 79916

Commander
HQ, TRADOC Combined Arms Test Activity
ATTN: ATCAT-OP-Q (CPT Henry C. Cobb, Jr.)
Fort Hood, TX 76544

Commander
HQ, TRADOC Combined Arms Test Activity
ATTN: ATCAT-SCI (Dr. Darrell W. Collier)
Fort Hood, TX 76544

Commander
US Army Dugway Proving Ground
ATTN: STEDP-MT-DA-L
Dugway, UT 84022

Commander
US Army Dugway Proving Ground
ATTN: STEDP-MT-DA-M (Mr. Paul E. Carlson)
Dugway, UT 84022

Commander
US Army Dugway Proving Ground
ATTN: STEDP-MT-DA-T (Mr. John Trethewey)
Dugway, UT 84022

Commander
US Army Dugway Proving Ground
ATTN: STEDP-MT-DA-T (Mr. William Peterson)
Dugway, UT 84022

Defense Documentation Center

ATTN: DDC-TCA
Cameron Station Bldg 5
Alexandria, VA 22314
12

Ballistic Missile Defense Program Office
ATTN: DACS-BMT (Colonel Harry F. Ennis)
5001 Eisenhower Avenue
Alexandria, VA 22333

Defense Technical Information Center
ATTN: DDA-2 (Mr. James E. Shafer)
Cameron Station, Bldg 5
Alexandria, VA 22314

Commander
US Army Materiel Development
& Readiness Command
ATTN: DRCBSI-EE (Mr. Albert Giambalvo)
5001 Eisenhower Avenue
Alexandria, VA 22333

Commander
US Army Materiel Development
& Readiness Command
ATTN: DRCLDC (Mr. James Bender)
5001 Eisenhower Avenue
Alexandria, VA 22333

Defense Advanced Rsch Projects Agency
ATTN: Steve Zakanyez
1400 Wilson Blvd
Arlington, VA 22209

Defense Advanced Rsch Projects Agency
ATTN: Dr. James Tegnalia
1400 Wilson Blvd
Arlington, VA 22209

Institute for Defense Analyses
ATTN: Mr. Lucien M. Biberman
400 Army-Navy Drive
Arlington, VA 22202

Institute for Defense Analyses
ATTN: Dr. Ernest Bauer
400 Army-Navy Drive
Arlington, VA 22202

Institute for Defense Analyses
ATTN: Dr. Hans G. Wolfhard
400 Army-Navy Drive
Arlington, VA 22202

System Planning Corporation
ATTN: Mr. Daniel Friedman
1500 Wilson Boulevard
Arlington, VA 22209

System Planning Corporation
ATTN: COL Hank Shelton
1500 Wilson Boulevard
Arlington, VA 22209

US Army Intelligence & Security Command
ATTN: Edwin Speakman, Scientific Advisor
Arlington Hall Station
Arlington, VA 22212

Commander
US Army Operational Test
& Evaluation Agency
ATTN: CSTE-ED (Mr. Floyd I. Hill)
5600 Columbia Pike
Falls Church, VA 22041

Commander and Director
US Army Engineer Topographic Laboratories
ATTN: ETL-GS-A (Mr. Thomas Neidringhaus)
Fort Belvoir, VA 22060

Director
US Army Night Vision &
Electro-Optics Laboratory
ATTN: DELNV-L (Dr. Rudolf G. Buser)
Fort Belvoir, VA 22060

Director
US Army Night Vision &
Electro-Optics Laboratory
ATTN: DELNV-L (Dr. Robert S. Rodhe)
Fort Belvoir, VA 22060

Director
US Army Night Vision &
Electro-Optics Laboratory
ATTN: DELNV-VI (Mr. Joseph R. Moulton)
Fort Belvoir, VA 22060

Director
US Army Night Vision &
Electro-Optics Laboratory
ATTN: DELNV-VI (Luanne P. Obert)
Fort Belvoir, VA 22060

Director
US Army Night Vision
& Electro-Optics Laboratory
ATTN: DELNV-VI (Mr. Thomas W. Cassidy)
Fort Belvoir, VA 22060

Director
US Army Night Vision &
Electro-Optics Laboratory
ATTN: DELNV-VI (Mr. Richard J. Bergemann)
Fort Belvoir, VA 22060

Director
US Army Night Vision &
Electro-Optics Laboratory
ATTN: DELNV-VI (Dr. James A. Ratches)
Fort Belvoir, VA 22060

Commander
US Army Training & Doctrine Command
ATTN: ATCD-AN
Fort Monroe, VA 23651

Commander
US Army Training & Doctrine Command
ATTN: ATCD-AN-M
Fort Monroe, VA 23651

Commander
US Army Training & Doctrine Command
ATTN: ATCD-F-A (Mr. Chris O'Connor, Jr.)
Fort Monroe, VA 23651

Commander
US Army Training & Doctrine Command
ATTN: ATCD-IE-R (Mr. David M. Ingram)
Fort Monroe, VA 23651

Commander
US Army Training & Doctrine Command
ATTN: ATCD-M-I/ATCD-M-A
Fort Monroe, VA 23651

Commander
US Army Training & Doctrine Command
ATTN: ATDOC-TA (Dr. Marvin P. Pastel)
Fort Monroe, VA 23651

Department of the Air Force
OL-I, AWS
Fort Monroe, VA 23651

Department of the Air Force
HQS 5 Weather Wing (MAC)
ATTN: 5 WW/DN
Langley Air Force Base, VA 23655

Commander
US Army INSCOM/Quest Research Corporation
ATTN: Mr. Donald Wilmot
6845 Elm Street, Suite 407
McLean, VA 22101

General Research Corporation
ATTN: Dr. Ralph Zirkind
7655 Old Springhouse Road
McLean, VA 22102

Science Applications, Inc.
8400 Westpark Drive
ATTN: Dr. John E. Cockayne
McLean, VA 22102

US Army Nuclear & Chemical Agency
ATTN: MONA-WE (Dr. John A. Berberet)
7500 Backlick Road, Bldg 2073
Springfield, VA 22150

Director
US Army Signals Warfare Laboratory
ATTN: DELSW-EA (Mr. Douglas Harkleroad)
Vint Hill Farms Station
Warrenton, VA 22186

Director
US Army Signals Warfare Laboratory
ATTN: DELSW-OS (Dr. Royal H. Burkhardt)
Vint Hill Farms Station
Warrenton, VA 22186

Commander
US Army Cold Regions Test Center
ATTN: STECR-TD (Mr. Jerold Barger)
APO Seattle, WA 98733

HQDA (SAUS-OR/Hunter M. Woodall, Jr./
Dr. Herbert K. Fallin)
Rm 2E 614, Pentagon
Washington, DC 20301

COL Elbert W. Friday, Jr.
OUSORE
Rm 3D 129, Pentagon
Washington, DC 20301

Defense Communications Agency
Technical Library Center
Code 222
Washington, DC 20305

Director
Defense Nuclear Agency
ATTN: Technical Library (Mrs. Betty Fox)
Washington, DC 20305

Director
Defense Nuclear Agency
ATTN: RAAE (Dr. Carl Fitz)
Washington, DC 20305

Director
Defense Nuclear Agency
ATTN: SPAS (Mr. Donald J. Kohler)
Washington, DC 20305

Defense Intelligence Agency
ATTN: DT/AC (LTC Robert Poplawski)
Washington, DC 20301

HQDA (DAMA-ARZ-D/Dr. Verderame)
Washington, DC 20310

HQDA (DAMI-ISP/Mr. Beck)
Washington, DC 20310

Department of the Army
Deputy Chief of Staff for
Operations and Plans
ATTN: DAMO-RQ
Washington, DC 20310

Department of the Army
Director of Telecommunications and
Command and Control
ATTN: DAMO-TCZ
Washington, DC 20310

Department of the Army
Assistant Chief of Staff for Intelligence
ATTN: DAMI-TS
Washington, DC 20310

HQDA (DAEN-RDM/Dr. de Percin)
Casimir Pulaski Building
20 Massachusetts Avenue
Room 6203
Washington, DC 20314

National Science Foundation
Division of Atmospheric Sciences
ATTN: Dr. Eugene W. Bierly
1800 G. Street, N.W.
Washington, DC 20550

Director
Naval Research Laboratory
ATTN: Code 4320 (Dr. Lothar H. Ruhnke)
Washington, DC 20375

Commanding Officer
Naval Research Laboratory
ATTN: Code 6009 (Dr. John MacCallum, Jr.)
Washington, DC 20375

Commanding Officer
Naval Research Laboratory
ATTN: Code 6530 (Mr. Raymond A. Patten)
Washington, DC 20375

Commanding Officer
Naval Research Laboratory
ATTN: Code 6533 (Dr. James A. Dowling)
Washington, DC 20375

ATMOSPHERIC SCIENCES RESEARCH REPORTS

1. Lindberg, J. D. "An Improvement to a Method for Measuring the Absorption Coefficient of Atmospheric Dust and other Strongly Absorbing Powders," ECOM-5565, July 1975.
2. Avara, Elton P., "Mesoscale Wind Shears Derived from Thermal Winds," ECOM-5566, July 1975.
3. Gomez, Richard B., and Joseph H. Pierluissi, "Incomplete Gamma Function Approximation for King's Strong-Line Transmittance Model," ECOM-5567, July 1975.
4. Blanco, A. J., and B. F. Engebos, "Ballistic Wind Weighting Functions for Tank Projectiles," ECOM-5568, August 1975.
5. Taylor, Fredrick J., Jack Smith, and Thomas H. Pries, "Crosswind Measurements through Pattern Recognition Techniques," ECOM-5569, July 1975.
6. Walters, D. L., "Crosswind Weighting Functions for Direct-Fire Projectiles," ECOM-5570, August 1975.
7. Duncan, Louis D., "An Improved Algorithm for the Iterated Minimal Information Solution for Remote Sounding of Temperature," ECOM-5571, August 1975.
8. Robbiani, Raymond L., "Tactical Field Demonstration of Mobile Weather Radar Set AN/TPS-41 at Fort Rucker, Alabama," ECOM-5572, August 1975.
9. Miers, B., G. Blackman, D. Langer, and N. Lorimier, "Analysis of SMS/GOES Film Data," ECOM-5573, September 1975.
10. Manquero, Carlos, Louis Duncan, and Rufus Bruce, "An Indication from Satellite Measurements of Atmospheric CO₂ Variability," ECOM-5574, September 1975.
11. Petracca, Carmine, and James D. Lindberg, "Installation and Operation of an Atmospheric Particulate Collector," ECOM-5575, September 1975.
12. Avara, Elton P., and George Alexander, "Empirical Investigation of Three Iterative Methods for Inverting the Radiative Transfer Equation," ECOM-5576, October 1975.
13. Alexander, George D., "A Digital Data Acquisition Interface for the SMS Direct Readout Ground Station - Concept and Preliminary Design," ECOM-5577, October 1975.
14. Cantor, Israel, "Enhancement of Point Source Thermal Radiation Under Clouds in a Nonattenuating Medium," ECOM-5578, October 1975.

15. Norton, Colburn, and Glenn Hoidale, "The Diurnal Variation of Mixing Height by Month over White Sands Missile Range, NM," ECOM-5579, November 1975.
16. Avara, Elton P., "On the Spectrum Analysis of Binary Data," ECOM-5580, November 1975.
17. Taylor, Fredrick J., Thomas H. Pries, and Chao-Huan Huang, "Optimal Wind Velocity Estimation," ECOM-5581, December 1975.
18. Avara, Elton P., "Some Effects of Autocorrelated and Cross-Correlated Noise on the Analysis of Variance," ECOM-5582, December 1975.
19. Gillespie, Patti S., R. L. Armstrong, and Kenneth O. White, "The Spectral Characteristics and Atmospheric CO₂ Absorption of the Ho⁺³:YLF Laser at 2.05 μ m," ECOM-5583, December 1975.
20. Novlan, David J., "An Empirical Method of Forecasting Thunderstorms for the White Sands Missile Range," ECOM-5584, February 1976.
21. Avara, Elton P., "Randomization Effects in Hypothesis Testing with Autocorrelated Noise," ECOM-5585, February 1976.
22. Watkins, Wendell R., "Improvements in Long Path Absorption Cell Measurement," ECOM-5586, March 1976.
23. Thomas, Joe, George D. Alexander, and Marvin Dubbin, "SATTEL - An Army Dedicated Meteorological Telemetry System," ECOM-5587, March 1976.
24. Kennedy, Bruce W., and Delbert Bynum, "Army User Test Program for the RDT&E-XM-75 Meteorological Rocket," ECOM-5588, April 1976.
25. Barnett, Kenneth M., "A Description of the Artillery Meteorological Comparisons at White Sands Missile Range, October 1974 - December 1974 ('PASS' - Prototype Artillery [Meteorological] Subsystem)," ECOM-5589, April 1976.
26. Miller, Walter B., "Preliminary Analysis of Fall-of-Shot From Project 'PASS'," ECOM-5590, April 1976.
27. Avara, Elton P., "Error Analysis of Minimum Information and Smith's Direct Methods for Inverting the Radiative Transfer Equation," ECOM-5591, April 1976.
28. Yee, Young P., James D. Horn, and George Alexander, "Synoptic Thermal Wind Calculations from Radiosonde Observations Over the Southwestern United States," ECOM-5592, May 1976.

29. Duncan, Louis D., and Mary Ann Seagraves, "Applications of Empirical Corrections to NOAA-4 VTPR Observations," ECOM-5593, May 1976.
30. Miers, Bruce T., and Steve Weaver, "Applications of Meteorological Satellite Data to Weather Sensitive Army Operations," ECOM-5594, May 1976.
31. Sharenow, Moses, "Redesign and Improvement of Balloon ML-566," ECOM-5595, June 1976.
32. Hansen, Frank V., "The Depth of the Surface Boundary Layer," ECOM-5596, June 1976.
33. Pinnick, R. G., and E. B. Stenmark, "Response Calculations for a Commerical Light-Scattering Aerosol Counter," ECOM-5597, July 1976.
34. Mason, J., and G. B. Hoidale, "Visibility as an Estimator of Infrared Transmittance," ECOM-5598, July 1976.
35. Bruce, Rufus E., Louis D. Duncan, and Joseph H. Pierluissi, "Experimental Study of the Relationship Between Radiosonde Temperatures and Radiometric-Area Temperatures," ECOM-5599, August 1976.
36. Duncan, Louis D., "Stratospheric Wind Shear Computed from Satellite Thermal Sounder Measurements," ECOM-5800, September 1976.
37. Taylor, F., P. Mohan, P. Joseph, and T. Pries, "An All Digital Automated Wind Measurement System," ECOM-5801, September 1976.
38. Bruce, Charles, "Development of Spectrophones for CW and Pulsed Radiation Sources," ECOM-5802, September 1976.
39. Duncan, Louis D., and Mary Ann Seagraves, "Another Method for Estimating Clear Column Radiances," ECOM-5803, October 1976.
40. Blanco, Abel J., and Larry E. Taylor, "Artillery Meteorological Analysis of Project Pass," ECOM-5804, October 1976.
41. Miller, Walter, and Bernard Engebos, "A Mathematical Structure for Refinement of Sound Ranging Estimates," ECOM-5805, November 1976.
42. Gillespie, James B., and James D. Lindberg, "A Method to Obtain Diffuse Reflectance Measurements from 1.0 and 3.0 μ m Using a Cary 17I Spectrophotometer," ECOM-5806, November 1976.
43. Rubio, Roberto, and Robert O. Olsen, "A Study of the Effects of Temperature Variations on Radio Wave Absorption," ECOM-5807, November 1976.

44. Ballard, Harold N., "Temperature Measurements in the Stratosphere from Balloon-Borne Instrument Platforms, 1968-1975," ECOM-5808, December 1976.
45. Monahan, H. H., "An Approach to the Short-Range Prediction of Early Morning Radiation Fog," ECOM-5809, January 1977.
46. Engebos, Bernard Francis, "Introduction to Multiple State Multiple Action Decision Theory and Its Relation to Mixing Structures," ECOM-5810, January 1977.
47. Low, Richard D. H., "Effects of Cloud Particles on Remote Sensing from Space in the 10-Micrometer Infrared Region," ECOM-5811, January 1977.
48. Bonner, Robert S., and R. Newton, "Application of the AN/GVS-5 Laser Rangefinder to Cloud Base Height Measurements," ECOM-5812, February 1977.
49. Rubio, Roberto, "Lidar Detection of Subvisible Reentry Vehicle Erosive Atmospheric Material," ECOM-5813, March 1977.
50. Low, Richard D. H., and J. D. Horn, "Mesoscale Determination of Cloud-Top Height: Problems and Solutions," ECOM-5814, March 1977.
51. Duncan, Louis D., and Mary Ann Seagraves, "Evaluation of the NOAA-4 VTPR Thermal Winds for Nuclear Fallout Predictions," ECOM-5815, March 1977.
52. Randhawa, Jagir S., M. Izquierdo, Carlos McDonald, and Zvi Salpeter, "Stratospheric Ozone Density as Measured by a Chemiluminescent Sensor During the Stratcom VI-A Flight," ECOM-5816, April 1977.
53. Rubio, Roberto, and Mike Izquierdo, "Measurements of Net Atmospheric Irradiance in the 0.7- to 2.8-Micrometer Infrared Region," ECOM-5817, May 1977.
54. Ballard, Harold N., Jose M. Serna, and Frank P. Hudson, Consultant for Chemical Kinetics, "Calculation of Selected Atmospheric Composition Parameters for the Mid-Latitude, September Stratosphere," ECOM-5818, May 1977.
55. Mitchell, J. D., R. S. Sagar, and R. O. Olsen, "Positive Ions in the Middle Atmosphere During Sunrise Conditions," ECOM-5819, May 1977.
56. White, Kenneth O., Wendell R. Watkins, Stuart A. Schleusener, and Ronald L. Johnson, "Solid-State Laser Wavelength Identification Using a Reference Absorber," ECOM-5820, June 1977.
57. Watkins, Wendell R., and Richard G. Dixon, "Automation of Long-Path Absorption Cell Measurements," ECOM-5821, June 1977.

58. Taylor, S. E., J. M. Davis, and J. B. Mason, "Analysis of Observed Soil Skin Moisture Effects on Reflectance," ECOM-5822, June 1977.
59. Duncan, Louis D., and Mary Ann Seagraves, "Fallout Predictions Computed from Satellite Derived Winds," ECOM-5823, June 1977.
60. Snider, D. E., D. G. Murcray, F. H. Murcray, and W. J. Williams, "Investigation of High-Altitude Enhanced Infrared Background Emissions," (U), SECRET, ECOM-5824, June 1977.
61. Dubbin, Marvin H., and Dennis Hall, "Synchronous Meteorological Satellite Direct Readout Ground System Digital Video Electronics," ECOM-5825, June 1977.
62. Miller, W., and B. Engebos, "A Preliminary Analysis of Two Sound Ranging Algorithms," ECOM-5826, July 1977.
63. Kennedy, Bruce W., and James K. Luers, "Ballistic Sphere Techniques for Measuring Atmospheric Parameters," ECOM-5827, July 1977.
64. Duncan, Louis D., "Zenith Angle Variation of Satellite Thermal Sounder Measurements," ECOM-5828, August 1977.
65. Hansen, Frank V., "The Critical Richardson Number," ECOM-5829, September 1977.
66. Ballard, Harold N., and Frank P. Hudson (Compilers), "Stratospheric Composition Balloon-Borne Experiment," ECOM-5830, October 1977.
67. Barr, William C., and Arnold C. Peterson, "Wind Measuring Accuracy Test of Meteorological Systems," ECOM-5831, November 1977.
68. Ethridge, G. A., and F. V. Hansen, "Atmospheric Diffusion: Similarity Theory and Empirical Derivations for Use in Boundary Layer Diffusion Problems," ECOM-5832, November 1977.
69. Low, Richard D. H., "The Internal Cloud Radiation Field and a Technique for Determining Cloud Blackness," ECOM-5833, December 1977.
70. Watkins, Wendell R., Kenneth O. White, Charles W. Bruce, Donald L. Walters, and James D. Lindberg, "Measurements Required for Prediction of High Energy Laser Transmission," ECOM-5834, December 1977.
71. Rubio, Robert, "Investigation of Abrupt Decreases in Atmospherically Backscattered Laser Energy," ECOM-5835, December 1977.
72. Monahan, H. H., and R. M. Cionco, "An Interpretative Review of Existing Capabilities for Measuring and Forecasting Selected Weather Variables (Emphasizing Remote Means)," ASL-TR-0001, January 1978.

73. Heaps, Melvin G., "The 1979 Solar Eclipse and Validation of D-Region Models," ASL-TR-0002, March 1978.
74. Jennings, S. G., and J. B. Gillespie, "M.I.E. Theory Sensitivity Studies - The Effects of Aerosol Complex Refractive Index and Size Distribution Variations on Extinction and Absorption Coefficients, Part II: Analysis of the Computational Results," ASL-TR-0003, March 1978.
75. White, Kenneth O., et al, "Water Vapor Continuum Absorption in the 3.5 μ m to 4.0 μ m Region," ASL-TR-0004, March 1978.
76. Olsen, Robert O., and Bruce W. Kennedy, "ABRES Pretest Atmospheric Measurements," ASL-TR-0005, April 1978.
77. Ballard, Harold N., Jose M. Serna, and Frank P. Hudson, "Calculation of Atmospheric Composition in the High Latitude September Stratosphere," ASL-TR-0006, May 1978.
78. Watkins, Wendell R., et al, "Water Vapor Absorption Coefficients at HF Laser Wavelengths," ASL-TR-0007, May 1978.
79. Hansen, Frank V., "The Growth and Prediction of Nocturnal Inversions," ASL-TR-0008, May 1978.
80. Samuel, Christine, Charles Bruce, and Ralph Brewer, "Spectrophone Analysis of Gas Samples Obtained at Field Site," ASL-TR-0009, June 1978.
81. Pinnick, R. G., et al., "Vertical Structure in Atmospheric Fog and Haze and its Effects on IR Extinction," ASL-TR-0010, July 1978.
82. Low, Richard D. H., Louis D. Duncan, and Richard B. Gomez, "The Microphysical Basis of Fog Optical Characterization," ASL-TR-0011, August 1978.
83. Heaps, Melvin G., "The Effect of a Solar Proton Event on the Minor Neutral Constituents of the Summer Polar Mesosphere," ASL-TR-0012, August 1978.
84. Mason, James B., "Light Attenuation in Falling Snow," ASL-TR-0013, August 1978.
85. Blanco, Abel J., "Long-Range Artillery Sound Ranging: 'PASS' Meteorological Application," ASL-TR-0014, September 1978.
86. Heaps, M. G., and F. E. Niles, "Modeling of Ion Chemistry of the D-Region: A Case Study Based Upon the 1966 Total Solar Eclipse," ASL-TR-0015, September 1978.

87. Jennings, S. G., and R. G. Pinnick, "Effects of Particulate Complex Refractive Index and Particle Size Distribution Variations on Atmospheric Extinction and Absorption for Visible Through Middle-Infrared Wavelengths," ASL-TR-0016, September 1978.
88. Watkins, Wendell R., Kenneth O. White, Lanny R. Bower, and Brian Z. Sojka, "Pressure Dependence of the Water Vapor Continuum Absorption in the 3.5- to 4.0-Micrometer Region," ASL-TR-0017, September 1978.
89. Miller, W. B., and B. F. Engebos, "Behavior of Four Sound Ranging Techniques in an Idealized Physical Environment," ASL-TR-0018, September 1978.
90. Gomez, Richard G., "Effectiveness Studies of the CBU-88/B Bomb, Cluster, Smoke Weapon," (U), CONFIDENTIAL ASL-TR-0019, September 1978.
91. Miller, August, Richard C. Shirkey, and Mary Ann Seagraves, "Calculation of Thermal Emission from Aerosols Using the Doubling Technique," ASL-TR-0020, November 1978.
92. Lindberg, James D., et al, "Measured Effects of Battlefield Dust and Smoke on Visible, Infrared, and Millimeter Wavelengths Propagation: A Preliminary Report on Dusty Infrared Test-I (DIRT-I)," ASL-TR-0021, January 1979.
93. Kennedy, Bruce W., Arthur Kinghorn, and B. R. Hixon, "Engineering Flight Tests of Range Meteorological Sounding System Radiosonde," ASL-TR-0022, February 1979.
94. Rubio, Roberto, and Don Hooek, "Microwave Effective Earth Radius Factor Variability at Wiesbaden and Balboa," ASL-TR-0023, February 1979.
95. Low, Richard D. H., "A Theoretical Investigation of Cloud/Fog Optical Properties and Their Spectral Correlations," ASL-TR-0024, February 1979.
96. Pinnick, R. G., and H. J. Auvermann, "Response Characteristics of Knollenberg Light-Scattering Aerosol Counters," ASL-TR-0025, February 1979.
97. Heaps, Melvin G., Robert O. Olsen, and Warren W. Berning, "Solar Eclipse 1979, Atmospheric Sciences Laboratory Program Overview," ASL-TR-0026, February 1979.
98. Blanco, Abel J., "Long-Range Artillery Sound Ranging: 'PASS' GR-8 Sound Ranging Data," ASL-TR-0027, March 1979.
99. Kennedy, Bruce W., and Jose M. Serna, "Meteorological Rocket Network System Reliability," ASL-TR-0028, March 1979.

100. Swingle, Donald M., "Effects of Arrival Time Errors in Weighted Range Equation Solutions for Linear Base Sound Ranging," ASL-TR-0029, April 1979.
101. Umstead, Robert K., Ricardo Pena, and Frank V. Hansen, "KWIK: An Algorithm for Calculating Munition Expenditures for Smoke Screening/Obscuration in Tactical Situations," ASL-TR-0030, April 1979.
102. D'Arcy, Edward M., "Accuracy Validation of the Modified Nike Hercules Radar," ASL-TR-0031, May 1979.
103. Rodriguez, Ruben, "Evaluation of the Passive Remote Crosswind Sensor," ASL-TR-0032, May 1979.
104. Barber, T. L., and R. Rodriguez, "Transit Time Lidar Measurement of Near-Surface Winds in the Atmosphere," ASL-TR-0033, May 1979.
105. Low, Richard D. H., Louis D. Duncan, and Y. Y. Roger R. Hsiao, "Micro-physical and Optical Properties of California Coastal Fogs at Fort Ord," ASL-TR-0034, June 1979.
106. Rodriguez, Ruben, and William J. Vechione, "Evaluation of the Saturation Resistant Crosswind Sensor," ASL-TR-0035, July 1979.
107. Ohmstede, William D., "The Dynamics of Material Layers," ASL-TR-0036, July 1979.
108. Pinnick, R. G., S. G. Jennings, Petr Chylek, and H. J. Auvermann, "Relationships between IR Extinction Absorption, and Liquid Water Content of Fogs," ASL-TR-0037, August 1979.
109. Rodriguez, Ruben, and William J. Vechione, "Performance Evaluation of the Optical Crosswind Profiler," ASL-TR-0038, August 1979.
110. Miers, Bruce T., "Precipitation Estimation Using Satellite Data," ASL-TR-0039, September 1979.
111. Dickson, David H., and Charles M. Sonnenschein, "Helicopter Remote Wind Sensor System Description," ASL-TR-0040, September 1979.
112. Heaps, Melvin G., and Joseph M. Heimerl, "Validation of the Dairchem Code, I: Quiet Midlatitude Conditions," ASL-TR-0041, September 1979.
113. Bonner, Robert S., and William J. Lentz, "The Visioceilometer: A Portable Cloud Height and Visibility Indicator," ASL-TR-0042, October 1979.
114. Cohn, Stephen L., "The Role of Atmospheric Sulfates in Battlefield Obscurations," ASL-TR-0043, October 1979.

115. Fawbush, E. J., et al, "Characterization of Atmospheric Conditions at the High Energy Laser System Test Facility (HELSTF), White Sands Missile Range, New Mexico, Part I, 24 March to 8 April 1977," ASL-TR-0044, November 1979.
116. Barber, Ted L., "Short-Time Mass Variation in Natural Atmospheric Dust," ASL-TR-0045, November 1979.
117. Low, Richard D. H., "Fog Evolution in the Visible and Infrared Spectral Regions and its Meaning in Optical Modeling," ASL-TR-0046, December 1979.
118. Duncan, Louis D., et al, "The Electro-Optical Systems Atmospheric Effects Library, Volume I: Technical Documentation," ASL-TR-0047, December 1979.
119. Shirkey, R. C., et al, "Interim E-O SAEL, Volume II, Users Manual," ASL-TR-0048, December 1979.
120. Kobayashi, H. K., "Atmospheric Effects on Millimeter Radio Waves," ASL-TR-0049, January 1980.
121. Seagraves, Mary Ann, and Louis D. Duncan, "An Analysis of Transmittances Measured Through Battlefield Dust Clouds," ASL-TR-0050, February 1980.
122. Dickson, David H., and Jon E. Ottesen, "Helicopter Remote Wind Sensor Flight Test," ASL-TR-0051, February 1980.
123. Pinnick, R. G., and S. G. Jennings, "Relationships Between Radiative Properties and Mass Content of Phosphoric Acid, HC, Petroleum Oil, and Sulfuric Acid Military Smokes," ASL-TR-0052, April 1980.
124. Hinds, B. D., and J. B. Gillespie, "Optical Characterization of Atmospheric Particulates on San Nicolas Island, California," ASL-TR-0053, April 1980.
125. Miers, Bruce T., "Precipitation Estimation for Military Hydrology," ASL-TR-0054, April 1980.
126. Stenmark, Ernest B., "Objective Quality Control of Artillery Computer Meteorological Messages," ASL-TR-0055, April 1980.
127. Duncan, Louis D., and Richard D. H. Low, "Bimodal Size Distribution Models for Fogs at Meppen, Germany," ASL-TR-0056, April 1980.
128. Olsen, Robert O., and Jagir S. Randhawa, "The Influence of Atmospheric Dynamics on Ozone and Temperature Structure," ASL-TR-0057, May 1980.

129. Kennedy, Bruce W., et al, "Dusty Infrared Test-II (DIRT-II) Program," ASL-TR-0058, May 1980.
130. Heaps, Melvin G., Robert O. Olsen, Warren Berning, John Cross, and Arthur Gilcrease, "1979 Solar Eclipse, Part I - Atmospheric Sciences Laboratory Field Program Summary," ASL-TR-0059, May 1980
131. Miller, Walter B., "User's Guide for Passive Target Acquisition Program Two (PTAP-2)," ASL-TR-0060, June 1980.
132. Holt, E. H., editor, "Atmospheric Data Requirements for Battlefield Obscuration Applications," ASL-TR-0061, June 1980.
133. Shirkey, Richard C., August Miller, George H. Goedecke, and Yugal Behl, "Single Scattering Code AGAUSX: Theory, Applications, Comparisons, and Listing," ASL-TR-0062, July 1980.
134. Sojka, Brian Z., and Kenneth O. White, "Evaluation of Specialized Photoacoustic Absorption Chambers for Near-Millimeter Wave (NMMW) Propagation Measurements," ASL-TR-0063, August 1980.
135. Bruce, Charles W., Young Paul Yee, and S. G. Jennings, "In Situ Measurement of the Ratio of Aerosol Absorption to Extinction Coefficient," ASL-TR-0064, August 1980.
136. Yee, Young Paul, Charles W. Bruce, and Ralph J. Brewer, "Gaseous/Particulate Absorption Studies at WSMR using Laser Sourced Spectrophones," ASL-TR-0065, June 1980.
137. Lindberg, James D., Radon B. Loveland, Melvin Heaps, James B. Gillespie, and Andrew F. Lewis, "Battlefield Dust and Atmospheric Characterization Measurements During West German Summertime Conditions in Support of Grafenwohr Tests," ASL-TR-0066, September 1980.
138. Vechione, W. J., "Evaluation of the Environmental Instruments, Incorporated Series 200 Dual Component Wind Set," ASL-TR-0067, September 1980.
139. Bruce, C. W., Y. P. Yee, B. D. Hinds, R. G. Pinnick, R. J. Brewer, and J. Minjares, "Initial Field Measurements of Atmospheric Absorption at 9 μ m to 11 μ m Wavelengths," ASL-TR-0068, October 1980.
140. Heaps, M. G., R. O. Olsen, K. D. Baker, D. A. Burt, L. C. Howlett, L. L. Jensen, E. F. Pound, and G. D. Allred, "1979 Solar Eclipse: Part II Initial Results for Ionization Sources, Electron Density, and Minor Neutral Constituents," ASL-TR-0069, October 1980.
141. Low, Richard D. H., "One-Dimensional Cloud Microphysical Models for Central Europe and their Optical Properties," ASL-TR-0070, October 1980.

142. Duncan, Louis D., James D. Lindberg, and Radon B. Loveland, "An Empirical Model of the Vertical Structure of German Fogs," ASL-TR-0071, November 1980.
143. Duncan, Louis D., "EOSAEL 80, Volume I, Technical Documentation," ASL-TR-0072, January 1981.
144. Shirkey, R. C., and S. G. O'Brien, "EOSAEL 80, Volume II, Users Manual," ASL-TR-0073, January 1981.
145. Bruce, C. W., "Characterization of Aerosol Nonlinear Effects on a High-Power CO₂ Laser Beam," ASL-TR-0074, February 1981.
146. Duncan, Louis D., and James D. Lindberg, "Air Mass Considerations in Fog Optical Modeling," ASL-TR-0075, February 1981.
147. Kunkel, Kenneth E., "Evaluation of a Tethered Kite Anemometer," ASL-TR-0076, February 1981.
148. Kunkel, K. E., et al, "Characterization of Atmospheric Conditions at the High Energy Laser System Test Facility (HELSTF) White Sands Missile Range, New Mexico, August 1977 to October 1978, Part II, Optical Turbulence, Wind, Water Vapor Pressure, Temperature," ASL-TR-0077, February 1981.
149. Miers, Bruce T., "Weather Scenarios for Central Germany," ASL-TR-0078, February 1981.
150. Cogan, James L., "Sensitivity Analysis of a Mesoscale Moisture Model," ASL-TR-0079, March 1981.
151. Brewer, R. J., C. W. Bruce, and J. L. Mater, "Optoacoustic Spectroscopy of C₂H₂ at the 9 μ m and 10 μ m C¹⁸O₂ Laser Wavelengths," ASL-TR-0080, March 1981.
152. Swingle, Donald M., "Reducible Errors in the Artillery Sound Ranging Solution, Part I: The Curvature Correction" (U), SECRET, ASL-TR-0081, April 1981.
153. Miller, Walter B., "The Existence and Implications of a Fundamental System of Linear Equations in Sound Ranging" (U), SECRET, ASL-TR-0082, April 1981.
154. Bruce, Dorothy, Charles W. Bruce, and Young Paul Yee, "Experimentally Determined Relationship Between Extinction and Liquid Water Content," ASL-TR-0083, April 1981.
155. Seagraves, Mary Ann, "Visible and Infrared Obscuration Effects of Ice Fog," ASL-TR-0084, May 1981.

156. Watkins, Wendell R., and Kenneth O. White, "Wedge Absorption Remote Sensor," ASL-TR-0085, May 1981.
157. Watkins, Wendell R., Kenneth O. White, and Laura J. Crow, "Turbulence Effects on Open Air Multipaths," ASL-TR-0086, May 1981.
158. Blanco, Abel J., "Extending Application of the Artillery Computer Meteorological Message," ASL-TR-0087, May 1981.
159. Heaps, M. G., D. W. Hooch, R. O. Olsen, B. F. Engebos, and R. Rubio, "High Frequency Position Location: An Assessment of Limitations and Potential Improvements," ASL-TR-0088, May 1981.
160. Watkins, Wendell R., and Kenneth O. White, "Laboratory Facility for Measurement of Hot Gaseous Plume Radiative Transfer," ASL-TR-0089, June 1981.
161. Heaps, M. G., "Dust Cloud Models: Sensitivity of Calculated Transmittances to Variations in Input Parameters," ASL-TR-0090, June 1981.
162. Seagraves, Mary Ann, "Some Optical Properties of Blowing Snow," ASL-TR-0091, June 1981.
163. Kobayashi, Herbert K., "Effect of Hail, Snow, and Melting Hydrometeors on Millimeter Radio Waves," ASL-TR-0092, July 1981.
164. Cogan, James L., "Techniques for the Computation of Wind, Ceiling, and Extinction Coefficient Using Currently Acquired RPV Data," ASL-TR-0093, July 1981.
165. Miller, Walter B., and Bernard F. Engebos, "On the Possibility of Improved Estimates for Effective Wind and Temperature," (U), SECRET, ASL-TR-0094, August 1981.
166. Heaps, Melvin G., "The Effect of Ionospheric Variability on the Accuracy of High Frequency Position Location," ASL-TR-0095, August 1981.
167. Sutherland, Robert A., Donald W. Hooch, and Richard B. Gomez, "An Objective Summary of US Army Electro-Optical Modeling and Field Testing in an Obscuring Environment," ASL-TR-0096, October 1981.
168. Pinnick, R. G., et al, "Backscatter and Extinction in Water Clouds," ASL-TR-0097, October 1981.
169. Cole, Henry P., and Melvin G. Heaps, "Properties of Dust as an Electron and Ion Attachment Site for Use in D Region Ion Chemistry," ASL-TR-0098, October 1981.

170. Spellicy, Robert L., Laura J. Crow, and Kenneth O. White, "Water Vapor Absorption Coefficients at HF Laser Wavelengths Part II: Development of the Measurement System and Measurements at Simulated Altitudes to 10 KM," ASL-TR-0099, November 1981.
171. Cohn, Stephen L., "Transport and Diffusion Solutions for Obscuration Using the XM-825 Smoke Munition," ASL-TR-0100, November 1981.
172. Pinnick, R. G., D. M. Garvey, and L. D. Duncan, "Calibration of Knollenberg FSSP Light-Scattering Counters for Measurement of Cloud Droplets," ASL-TR-0101, December 1981.
173. Cohn, Stephen L. and Ricardo Pena, "Munition Expenditure Model Verification: KWIK Phase I," ASL-TR-0102, December 1981.
174. Blackman, George R., "Cloud Geometry Analysis of the Smoke Week III Obscuration Trials," ASL-TR-0103, January 1982.
175. Sutherland, R. A., and D. W. Hooch, "An Improved Smoke Obscuration Model ACT II: Part 1 Theory," ASL-TR-0104, January 1982.
176. Lentz, W. J., "The Visioceilometer: A Portable Visibility and Cloud Ceiling Height Lidar," ASL-TR-0105, January 1982.
177. Kennedy, Bruce W., "An Analysis of Craters Produced by Artillery Munitions During Dusty Infrared Test Series," ASL-TR-0106, January 1982.
178. Steinhoff, R. G., "Program Listings for EOSAEL 80-B and Ancillary Codes AGAUS and FLASH," ASL-TR-0107, February 1982.

DATE
ILME